Treasure:
In search of the Golden Horse

THE SOLUTION

Version 1 (2023-05-11)
Version 2 (2023-12-04)
by J.P. Labonté
@Labgo

Version 2 modifications from Version 1 are enclosed within light purple bands in the text. Chapters FORTUNE and ROAD, making up The Journey, were modified extensively as new solutions to puzzles involving previously untouched chapters (TMWBG, THE SIGNAL, LODGE) allowed the derivation of sentence selectors for these two chapters, streamlining the process and confirming the Journey's identification of states made in Version 1.

(Slight corrections made in the text were not flagged)

INTRODUCTION

This paper provides a solution to a famous puzzle published 39 years ago (©1984 Intravision Inc.). This solution gives the precise location of the golden horse in the form of longitude and latitude (degrees, minutes, seconds). This is a very sophisticated puzzle, a tribute to the talent of its author, the puzzle master Paul Hoffman (Dr. Crypton). This fairy tale (based on a story by Sheldon Renan and David Wyles, illustrated by J.F. Podevin) is not only beautifully constructed, the text is also highly self-referential. The body of the text contains instructions on how to modify itself and extract from it the relevant information, often large amounts of text. There is no overarching rule to apply to all chapters. Instead, most chapters have their own puzzles to solve. Each of these chapters contribute a piece of the machinery that, when correctly assembled, allows interactions between the pieces to generate the coordinates we are looking for.

The challenge is therefore to identify the relevant text and how to manipulate it according to rules suggested to by the clues in each chapter. The geographic location of the golden horse is not to be

guessed by interpreting isolated clues that may point to the correct general geographic location. Although such clues are found in the book, these are no substitute for a process that blindly manipulates large amounts of text and finally, almost magically, produces the exact coordinates.

The solution does not require any advanced cryptographic knowledge. Everything can be done with a pen and paper. However, manipulating large amounts of text in specific ways is prone to mistakes. For this reason, *Mathematica* code was used in many instances to process the text according to specific rules. This ensures accuracy as long as the text is accurately copied to start with. From experience, it is easier to copy text accurately than it is to, for example, count the number of occurrences of specific letters in a body of text, something *Mathematica* can easily do.

Short history and references

The following is the trailer for the movie version of the book (available on tape and videodiscs - these two versions are slightly different). It includes an interview with Sheldon Renan:

https://www.youtube.com/watch?v=bjn--BUnniE

The following is the full movie:

https://www.youtube.com/watch?v=ymec2dg4O8A

The contest ran for 5 years (until midnight May 26, 1989) but the golden horse was not found and ended up being given to charity. The organizers never revealed the solution. Two men (Nick Boone and Anthony Castaneda) claimed to have solved the puzzle after the deadline. Here is an article in the Washington Post about their story:

https://www.washingtonpost.com/archive/lifestyle/1989/11/30/two-find-treasure-too-late/d62c5c8b-bb02-4027-acba-f20f12e64d47/

And here are two videos (Part 1 and Part 2) telling their full story:

https://www.youtube.com/watch?v=3V3OTePAjYw

https://www.youtube.com/watch?v=AulcXQO1YZw

Although Boone and Castaneda found the location, many believed that they did not find the intended solution or path, but instead relied on secondary clues also present in the book. This is akin to police detectives relying on weak circumstantial evidence to apprehend a suspect. They would never get a conviction in a court of law. The present paper offers instead a path that speaks to the incredible sophistication of this puzzle.

The starting point

The book cover

When confronted with this 21-chapter text, it quickly becomes apparent that some way to simplify the task is required. There must be a way to identify the starting point or the chapters to focus on. The book cover is a natural place to look for this information. Its rich imagery, centered on a maze, illustrates the complexity of the task at hands. And, as expected, it also contains the three most important clues to get you started:

- the number 7
- the number 31
- a page curled on itself

I came up with the numbers 7 and 31 years ago and tried to use this information in all sort of ways over the years ("the 7-31 solution"), but no satisfactory answer emerged... until now (2023). This is when I realized that the curly page was the missing clue. Note the word "Page" immediately above the curl. Only one page in the book shows a similar curl: **page 74, in COURTYARD** (chapter 20).

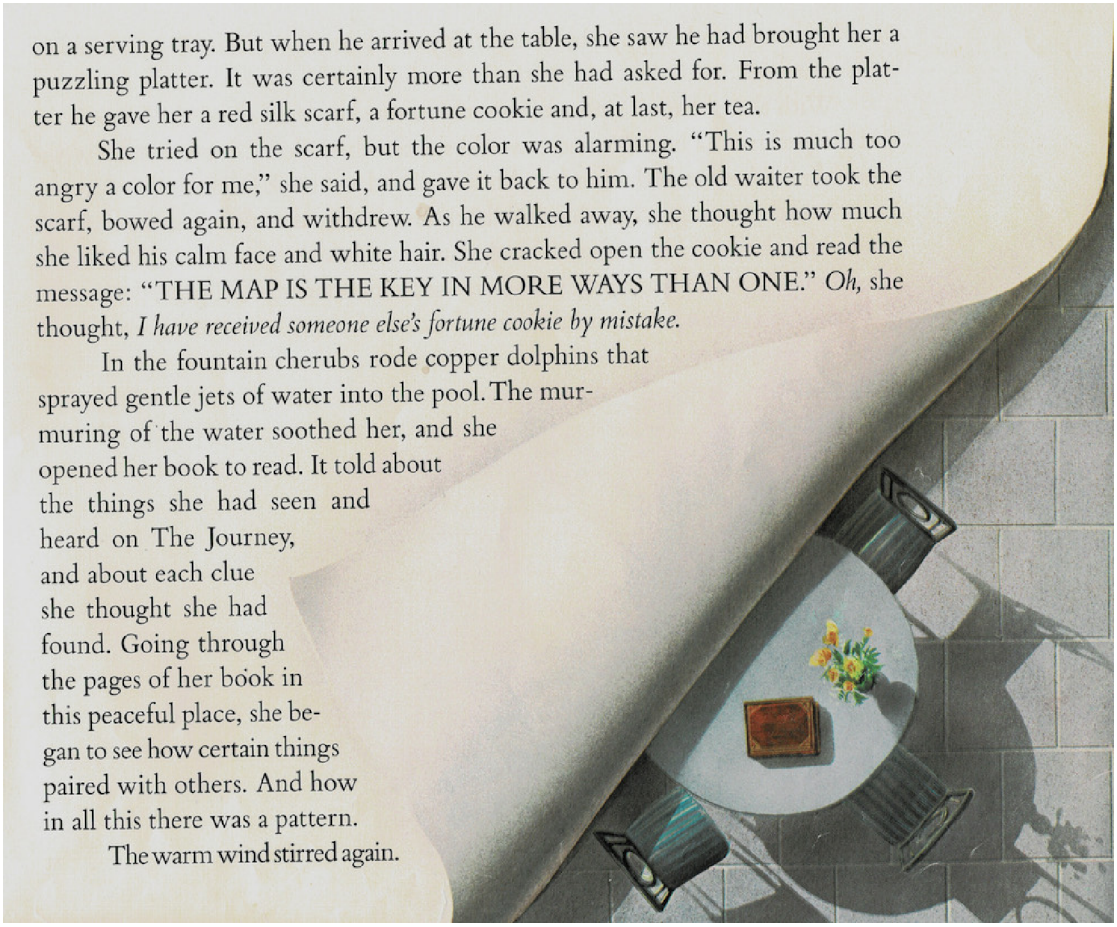
Chapter 20 - COURTYARD

Before starting this chapter, a few comments are in order:

- The purpose of this chapter is to provide a kind of guide to the main chapters involved in arriving at a solution.
  - understanding this chapter is therefore very useful, but not essential to the solution.
  - I actually completed my analysis of this chapter after I completed several other chapters. This involved a back and forth process to fine tune my interpretation of some clues that I did not see in the first place.
  - this chapter may contain more information than what I actually found.
- What do I mean by fine tuning? Here is the general approach for most chapters.
  - All the sentences in a chapter may not need to be processed. There is usually a way to select sentences that need to be processed further.
  - A **general rule** is then applied to the selected sentences (for example counting specific letters) to extract what is needed.
  - Then it is getting more complicated. **Specific rules** may have to be applied to **specific sentences**.
    - these specific rules are derived from the double meaning of the sentence itself
    - I have not found a way to identify which sentences are subjects to specific rules, hence the danger of missing some of these or applying some specific rule where it is not needed. In the context of the story you can often detect something odd or forced about these sentences.
    - After working on different chapters you get better at identifying these rules. This is why COURTYARD required some back and forth to fine tune its solution.
- I came to COURTYARD from the book cover with two prime numbers (7 and 31). My guiding idea was that I would discover what I called the **PRIME CHAPTERS**, those chapters that are most important in finding the solution. The chapters identified here may not be all there is to find, but they do lead to the exact coordinates. Other chapters (often images) are still useful in providing information that will help solve some of the puzzles.

Part 1: The table

Clues on the book cover get us to page 74, where we need to figure out how to use the prime numbers 7 and 31.



The image suggests that a table is required. Before going further we will use *Mathematica* to help us in manipulating the text and present things in a clean way. There is a lot to learn from this whole chapter and page 74 is only the beginning. *Mathematica* code blocks are shown with a yellow background. Some code blocks produce printed text as an output. These results appear in a block with a green background immediately following the code block .

```
(* The raw text of chapter COURTYARD. Note that two spaces
  have been added between text of pages 73-74 and 74-75  *)

courtyardRaw =
  "The night wind woke Amanda. She was exactly where she had fallen asleep: in
    Alice's lap, next to the tuxedo-clad rabbit. With her book she slid down
    to the ground and started to walk again. The night air was surprisingly
    warm. She left the forest and headed toward a glow in the sky. It was
    the light of a nearby town, a place where travelers rested from their
    journeys. It appeared on every map, but not all who traveled could
    find it. She had slept well into the night and when she arrived, the
    cobblestone streets were empty. On the edge of town, however, in a
    sweet-smelling grove of cypress, there was an old inn where the lamps
```

```
    still burned. Amanda followed the flickering flames to the center of
    the courtyard. A warm wind touched her cheek, and she felt strangely
    comfortable. She sat down at the first table she came to. A fountain
    bubbled nearby, casting reflected light across her table and onto the
    branches that hung above her. Along the far side of the courtyard,
    wooden fans whirled above an ancient bar. She saw an old waiter leaning
    against the bar. He was dressed in a black vest and tie. He stayed
    where he was, patiently waiting. He seemed to have been waiting for
    her for many years. “May I have some tea?” she asked. He bowed in
    answer and put a silver pot  on a serving tray. But when he arrived
    at the table, she saw he had brought her a puzzling platter. It was
    certainly more than she had asked for. From the platter he gave her
    a red silk scarf, a fortune cookie and, at last, her tea. She tried
    on the scarf, but the color was alarming. “This is much too angry a
    color for me,” she said, and give it back to him. The old waiter took
    the scarf, bowed again, and withdrew. As he walked away, she thought
    how much she liked his calm face and white hair. She cracked open the
    cookie and read the message: “THE MAP IS THE KEY IN MORE WAYS THAN
    ONE.” Oh, she thought, I have received someone else's fortune cookie
    by mistake. In the fountain cherubs rode copper dolphins that sprayed
    gentle jets of water into the pool. The murmuring of the water soothed
    her, and she opoened her book to read. It told about the things she
    had seen and heard on The Journey, and about each clue she thought
    she had found. Going through the pages of her book in this peaceful
    place, she began to see how certain things paired with others. And
    how in all this there was a pattern. The warm wind stirred again.
    How little it takes to be happy, she thought. If you know where to
    look, the path is always clearly marked. With that she took a pen and
    wrote just one last word. And she felt as if she escaped from a box.
    No more would she search endlessly. Now she knew she would find what
    she had intended. And she closed her book forever. In the morning,
    when the black bird flew overhead to find her, the book remained on
    the table like a ghost. But Amanda had left the courtyard. Never
    again would the black bird deceive her. The way could no longer be
    eclipsed. She had learned the most difficult of secret: how to see.";
```

```
(* The text on page 74, extracted from courtyardRaw with the
 help of the two spaces inserted earlier to separate pages *)

courtyardP74Raw = StringSplit[courtyardRaw, "  "][[2]]
```

```
on a serving tray. But when he arrived at the table, she saw he had brought her
  a puzzling platter. It was certainly more than she had asked for. From
  the platter he gave her a red silk scarf, a fortune cookie and, at last,
  her tea. She tried on the scarf, but the color was alarming. “This is
  much too angry a color for me,” she said, and give it back to him. The
  old waiter took the scarf, bowed again, and withdrew. As he walked
  away, she thought how much she liked his calm face and white hair. She
  cracked open the cookie and read the message: “THE MAP IS THE KEY IN
  MORE WAYS THAN ONE.” Oh, she thought, I have received someone else's
  fortune cookie by mistake. In the fountain cherubs rode copper dolphins
  that sprayed gentle jets of water into the pool. The murmuring of the
  water soothed her, and she opoened her book to read. It told about the
  things she had seen and heard on The Journey, and about each clue she
  thought she had found. Going through the pages of her book in this
  peaceful place, she began to see how certain things paired with others.
  And how in all this there was a pattern. The warm wind stirred again.
```

To facilitate the discussion, we will number the sentences in the chapter.

```
(* The full text of chapter COURTYARD, with numbered sentences *)

Column@Normal@(AssociationThread[
    Range[Length[TextSentences[courtyardRaw]]], TextSentences[courtyardRaw]])
```

```
1 → The night wind woke Amanda.
2 → She was exactly where she had fallen
   asleep: in Alice's lap, next to the tuxedo-clad rabbit.
3 → With her book she slid down to the ground and started to walk again.
4 → The night air was surprisingly warm.
5 → She left the forest and headed toward a glow in the sky.
6 → It was the light of a nearby town,
   a place where travelers rested from their journeys.
7 → It appeared on every map, but not all who traveled could find it.
8 → She had slept well into the night
   and when she arrived, the cobblestone streets were empty.
9 → On the edge of town, however, in a sweet-smelling grove
   of cypress, there was an old inn where the lamps still burned.
10 → Amanda followed the flickering flames to the center of the courtyard.
11 → A warm wind touched her cheek, and she felt strangely comfortable.
12 → She sat down at the first table she came to.
13 → A fountain bubbled nearby, casting reflected light
   across her table and onto the branches that hung above her.
14 →
 Along the far side of the courtyard, wooden fans whirled above an ancient bar.
```

```
15 → She saw an old waiter leaning against the bar.
16 → He was dressed in a black vest and tie.
17 → He stayed where he was, patiently waiting.
18 → He seemed to have been waiting for her for many years.
19 → “May I have some tea?” she asked.
20 → He bowed in answer and put a silver pot  on a serving tray.
21 → But when he arrived at the
   table, she saw he had brought her a puzzling platter.
22 → It was certainly more than she had asked for.
23 → From the platter he gave her a
   red silk scarf, a fortune cookie and, at last, her tea.
24 → She tried on the scarf, but the color was alarming.
25 → “This is much too angry a color for me,” she said, and give it back to him.
26 → The old waiter took the scarf, bowed again, and withdrew.
27 →
 As he walked away, she thought how much she liked his calm face and white hair.
28 → She cracked open the cookie and read
   the message: “THE MAP IS THE KEY IN MORE WAYS THAN ONE.”
29 → Oh, she thought, I have received someone else's fortune cookie by mistake.
30 → In the fountain cherubs rode copper
   dolphins that sprayed gentle jets of water into the pool.
31 → The murmuring of the water soothed her, and she opoened her book to read.
32 → It told about the things she had seen and heard
   on The Journey, and about each clue she thought she had found.
33 → Going through the pages of her book in this peaceful
   place, she began to see how certain things paired with others.
34 → And how in all this there was a pattern.
35 → The warm wind stirred again.
36 → How little it takes to be happy, she thought.
37 → If you know where to look, the path is always clearly marked.
38 → With that she took a pen and wrote just one last word.
39 → And she felt as if she escaped from a box.
40 → No more would she search endlessly.
41 → Now she knew she would find what she had intended.
42 → And she closed her book forever.
43 → In the morning, when the black bird flew overhead
   to find her, the book remained on the table like a ghost.
44 → But Amanda had left the courtyard.
45 → Never again would the black bird deceive her.
46 → The way could no longer be eclipsed.
47 → She had learned the most difficult of secret: how to see.
```

We may now get back to the task at hand. Can we use the prime numbers 7 and 31 to make a table with the words on page 74? This would require **217 words** (7x31)? How many words do we have on that page?

```
(* Number of raw words on page 74 *)

Length@TextWords@courtyardP74Raw
```

```
218
```

We have one too many word to fit in a 7x31 table. The culprit is found in sentence #29: **mistake**. We need to remove this word.

```
(* Remove "mistake" from courtyardP74Raw
 and give a list of the resulting 217 words *)

courtyardP74Words = TextWords@StringDelete[courtyardP74Raw, "mistake"]
```

```
{on, a, serving, tray, But, when, he, arrived, at, the, table, she, saw, he, had,
 brought, her, a, puzzling, platter, It, was, certainly, more, than, she, had,
 asked, for, From, the, platter, he, gave, her, a, red, silk, scarf, a, fortune,
 cookie, and, at, last, her, tea, She, tried, on, the, scarf, but, the, color,
 was, alarming, This, is, much, too, angry, a, color, for, me, she, said, and,
 give, it, back, to, him, The, old, waiter, took, the, scarf, bowed, again,
 and, withdrew, As, he, walked, away, she, thought, how, much, she, liked, his,
 calm, face, and, white, hair, She, cracked, open, the, cookie, and, read,
 the, message, THE, MAP, IS, THE, KEY, IN, MORE, WAYS, THAN, ONE, Oh, she,
 thought, I, have, received, someone, else's, fortune, cookie, by, In, the,
 fountain, cherubs, rode, copper, dolphins, that, sprayed, gentle, jets, of,
 water, into, the, pool, The, murmuring, of, the, water, soothed, her, and,
 she, opoened, her, book, to, read, It, told, about, the, things, she, had,
 seen, and, heard, on, The, Journey, and, about, each, clue, she, thought, she,
 had, found, Going, through, the, pages, of, her, book, in, this, peaceful,
 place, she, began, to, see, how, certain, things, paired, with, others, And,
 how, in, all, this, there, was, a, pattern, The, warm, wind, stirred, again}
```

We now have our 217 words to produce the table. But this is a circular table (as shown in the image), so we may have to rotate it in order to place the word we want at the beginning of the list. Sentence #12 provides the answer: "**first table**". The word "table" is currently at position 11, so we need to rotate the list to bring "table" at the beginning.

```
(* Make "table" first and create a 7x31 table *)

Grid[Partition[RotateLeft[courtyardP74Words, 10], 7],
 Frame → All, Background → White]
```

table	she	saw	he	had	brought	her
a	puzzling	platter	It	was	certainly	more
than	she	had	asked	for	From	the
platter	he	gave	her	a	red	silk
scarf	a	fortune	cookie	and	at	last
her	tea	She	tried	on	the	scarf
but	the	color	was	alarming	This	is
much	too	angry	a	color	for	me
she	said	and	give	it	back	to
him	The	old	waiter	took	the	scarf
bowed	again	and	withdrew	As	he	walked
away	she	thought	how	much	she	liked
his	calm	face	and	white	hair	She
cracked	open	the	cookie	and	read	the
message	THE	MAP	IS	THE	KEY	IN
MORE	WAYS	THAN	ONE	Oh	she	thought
I	have	received	someone	else's	fortune	cookie
by	In	the	fountain	cherubs	rode	copper
dolphins	that	sprayed	gentle	jets	of	water
into	the	pool	The	murmuring	of	the
water	soothed	her	and	she	opoened	her
book	to	read	It	told	about	the
things	she	had	seen	and	heard	on
The	Journey	and	about	each	clue	she
thought	she	had	found	Going	through	the
pages	of	her	book	in	this	peaceful
place	she	began	to	see	how	certain
things	paired	with	others	And	how	in
all	this	there	was	a	pattern	The
warm	wind	stirred	again	on	a	serving
tray	But	when	he	arrived	at	the

The rotation necessary to put the word “table” first had the side effect of putting the word “**the**” last. Sentence #23 tells us “at **last,** her tea”. The word **the** (with the “e” accented) is French for tea.

Part 2: sentence selection

What has to be done is not immediately obvious. We will first focus on the key **“THE MAP IS THE KEY IN MORE WAYS THAN ONE”** (Sentence #28), an obviously important clue (besides reminding us of the key role of the map in general). Given that we have 31 rows and that the key has 31 characters, a reasonable approach is to label each row with a letter from the key, in effect adding a key column to

the table.

```
(* Inserting the key column *)

Grid[
 Flatten@# & /@ Partition[Riffle[Characters@"THEMAPISTHEKEYINMOREWAYSTHANONE",
    Partition[RotateLeft[courtyardP74Words, 10], 7]], 2],
 Frame → All, Background → {1 → Lighter[Gray, 0.8], White}]
```

T	table	she	saw	he	had	brought	her
H	a	puzzling	platter	It	was	certainly	more
E	than	she	had	asked	for	From	the
M	platter	he	gave	her	a	red	silk
A	scarf	a	fortune	cookie	and	at	last
P	her	tea	She	tried	on	the	scarf
I	but	the	color	was	alarming	This	is
S	much	too	angry	a	color	for	me
T	she	said	and	give	it	back	to
H	him	The	old	waiter	took	the	scarf
E	bowed	again	and	withdrew	As	he	walked
K	away	she	thought	how	much	she	liked
E	his	calm	face	and	white	hair	She
Y	cracked	open	the	cookie	and	read	the
I	message	THE	MAP	IS	THE	KEY	IN
N	MORE	WAYS	THAN	ONE	Oh	she	thought
M	I	have	received	someone	else's	fortune	cookie
O	by	In	the	fountain	cherubs	rode	copper
R	dolphins	that	sprayed	gentle	jets	of	water
E	into	the	pool	The	murmuring	of	the
W	water	soothed	her	and	she	opoened	her
A	book	to	read	It	told	about	the
Y	things	she	had	seen	and	heard	on
S	The	Journey	and	about	each	clue	she
T	thought	she	had	found	Going	through	the
H	pages	of	her	book	in	this	peaceful
A	place	she	began	to	see	how	certain
N	things	paired	with	others	And	how	in
O	all	this	there	was	a	pattern	The
N	warm	wind	stirred	again	on	a	serving
E	tray	But	when	he	arrived	at	the

In sentence #25, Amanda returns the red scarf. Removing the word “scarf ” (which appears 3 times) would destroy the symmetry of our table. We should ask ourselves why an odd item like a scarf was given to Amanda to start with. It is because a scarf is a narrow piece of garment to be “**worn around** the neck or over the head”. The scarf includes all the words, from its first appearance to its last appearance. This covers a full six rows.

```
(* Highlighting the scarf *)

Grid[
 Flatten@# & /@ Partition[Riffle[Characters@"THEMAPISTHEKEYINMOREWAYSTHANONE",
    Partition[RotateLeft[courtyardP74Words, 10], 7]], 2],
 Frame → All, Background → {1 → Lighter[Gray, 0.8],
   {1 → White, 2 → White, 3 → White, 4 → White, 5 → LightRed, 6 → LightRed,
    7 → LightRed, 8 → LightRed, 9 → LightRed, 10 → LightRed, 11 → White, 12 → White,
    13 → White, 14 → White, 15 → White, 16 → White, 17 → White, 18 → White, 19 → White,
    20 → White, 21 → White, 22 → White, 23 → White, 24 → White, 25 → White,
    26 → White, 27 → White, 28 → White, 29 → White, 30 → White, 31 → White}}]
```

T	table	she	saw	he	had	brought	her
H	a	puzzling	platter	It	was	certainly	more
E	than	she	had	asked	for	From	the
M	platter	he	gave	her	a	red	silk
A	scarf	a	fortune	cookie	and	at	last
P	her	tea	She	tried	on	the	scarf
I	but	the	color	was	alarming	This	is
S	much	too	angry	a	color	for	me
T	she	said	and	give	it	back	to
H	him	The	old	waiter	took	the	scarf
E	bowed	again	and	withdrew	As	he	walked
K	away	she	thought	how	much	she	liked
E	his	calm	face	and	white	hair	She
Y	cracked	open	the	cookie	and	read	the
I	message	THE	MAP	IS	THE	KEY	IN
N	MORE	WAYS	THAN	ONE	Oh	she	thought
M	I	have	received	someone	else's	fortune	cookie
O	by	In	the	fountain	cherubs	rode	copper
R	dolphins	that	sprayed	gentle	jets	of	water
E	into	the	pool	The	murmuring	of	the
W	water	soothed	her	and	she	opoened	her
A	book	to	read	It	told	about	the
Y	things	she	had	seen	and	heard	on
S	The	Journey	and	about	each	clue	she
T	thought	she	had	found	Going	through	the
H	pages	of	her	book	in	this	peaceful
A	place	she	began	to	see	how	certain
N	things	paired	with	others	And	how	in
O	all	this	there	was	a	pattern	The
N	warm	wind	stirred	again	on	a	serving
E	tray	But	when	he	arrived	at	the

These are the letters in the key corresponding to the scarf area: **APISTH**

This is an anagram for **PATH IS**, sentence #37 does mention these exact words. But looking more

closely at the table we find that associated with the letters **IS** in the key are the words COLOR ALARMING and COLOR ANGRY in the table. These two rows should therefore not be considered. We are left with 4 rows and therefore the 4 letters **PATH** as our sentence selector. Any sentence containing the 4 letters **P A T H** in any order will be selected for the next step.

```
(* Selected sentences containing the letters P A T H *)

courtyardPathSentences =
  MapIndexed[If[StringContainsQ[#1, "P", IgnoreCase → True] &&
      StringContainsQ[#1, "A", IgnoreCase → True] && StringContainsQ[#1, "T",
       IgnoreCase → True] && StringContainsQ[#1, "H", IgnoreCase → True],
     Row[{#2[[1]], "- ", #1}], Nothing] &, TextSentences[courtyardRaw]];
Column@courtyardPathSentences
```

```
2- She was exactly where she had fallen
   asleep: in Alice's lap, next to the tuxedo-clad rabbit.
4- The night air was surprisingly warm.
6- It was the light of a nearby town,
   a place where travelers rested from their journeys.
7- It appeared on every map, but not all who traveled could find it.
8- She had slept well into the night
   and when she arrived, the cobblestone streets were empty.
9- On the edge of town, however, in a sweet-smelling grove
   of cypress, there was an old inn where the lamps still burned.
17- He stayed where he was, patiently waiting.
20- He bowed in answer and put a silver pot  on a serving tray.
21- But when he arrived at the
   table, she saw he had brought her a puzzling platter.
23- From the platter he gave her a
   red silk scarf, a fortune cookie and, at last, her tea.
28- She cracked open the cookie and read
   the message: “THE MAP IS THE KEY IN MORE WAYS THAN ONE.”
30- In the fountain cherubs rode copper
   dolphins that sprayed gentle jets of water into the pool.
31- The murmuring of the water soothed her, and she opoened her book to read.
33- Going through the pages of her book in this peaceful
   place, she began to see how certain things paired with others.
34- And how in all this there was a pattern.
36- How little it takes to be happy, she thought.
37- If you know where to look, the path is always clearly marked.
38- With that she took a pen and wrote just one last word.
39- And she felt as if she escaped from a box.
46- The way could no longer be eclipsed.
```

Part 3: operating on the selected sentences

We have 20 selected sentences, but 21 chapters. Given that the present chapter (COURTYARD) is included de facto in the important chapters, we will assume that COURTYARD is not counted and that these 20 sentences refer to the 20 chapters left. This is supported by sentence #44 telling us that "Amanda had left the courtyard". This is before we get to the twentieth sentence (46), that must therefore apply to chapter 21.

General rule

Sentence #38 tells us what we need: "**just one last word**". So we need to operate on the last word of each sentence. A typical operation with letters is to convert them to their numbers in the alphabet (a = 1, b = 2, c=3...). We will add the numbers associated with each letter of the last word in each sentence and get a value for that word. This value will determine if we have a **prime chapter**. How? The number must be prime! A prime number has "just one" as a factor (other than itself).

Sentence numbers are shown in a box with the chapter represented. Red entries indicate that the sentence did not result in a prime number. Black entries resulted in prime numbers and are therefore prime chapters.

2 – CREATION (ch. 1)

```
Total@LetterNumber@Characters["rabbit"]
```

```
52
```

Here we have to apply a **specific rule**. "rabbit" is to be interpreted as RAW BIT. So 52 must be expressed in binary:

```
BaseForm[52, 2]
```

110100_2

The number of bits set to one is **3, which is a prime number.** So CREATION is a prime chapter.

4 – THE MAN WITH BLACK GLOVES (ch .2)

```
Total@LetterNumber@Characters["warm"]
```

```
55
```

6 – KITES (ch. 3)

```
Total@LetterNumber@Characters["journeys"]
```

```
127
```

7 – CEMETERY (ch. 4)

```
Total@LetterNumber@Characters["it"]
```

```
29
```

8 – FORTUNE (ch .5)

```
Total@LetterNumber@Characters["empty"]
```

```
79
```

9 – FISH (ch .6)

```
Total@LetterNumber@Characters["burned"]
```

```
64
```

17 – WAITING (ch .7)

```
Total@LetterNumber@Characters["waiting"]
```

```
83
```

20 – WILDERNESS (ch .8)

```
Total@LetterNumber@Characters["tray"]
```

```
64
```

21 – MR. MAPS (ch .9)

```
Total@LetterNumber@Characters["platter"]
```

```
92
```

Here we have to apply a **specific rule**. We have a “puzzling platter” in sentence #21, which obviously suggests a puzzle. The puzzle is that the platter contains something else: a **cookie**, a **scarf** and **tea**. Using these names with or without “**platter**” gives a prime number. So MR. MAPS is also a prime chapter.

```
Total@LetterNumber@Characters["cookiescarftea"]
```

```
131
```

```
Total@LetterNumber@Characters["plattercookiescarftea"]
```

```
223
```

23 – THE SIGNAL (ch .10)

```
Total@LetterNumber@Characters["tea"]
```

```
26
```

28 – CAROUSEL (ch .11)

```
Total@LetterNumber@Characters["one"]
```

```
34
```

30 – MOUNTAIN (ch .12)

```
Total@LetterNumber@Characters["pool"]
```

```
58
```

31 – LODGE (ch .13)

```
Total@LetterNumber@Characters["read"]
```

```
28
```

33 – MEMORY (ch .14)

```
Total@LetterNumber@Characters["others"]
```

```
85
```

34 – RIVER (ch .15)

```
Total@LetterNumber@Characters["pattern"]
```

```
94
```

Here we have to apply a **specific rule**. “in all this there was a pattern”. We need to replace “pattern”. The pattern in the COURTYARD image is the floor. It is a “floor of **slate**” (as mentioned verbatim in chapter DIXIELAND). The pattern is in ALL THIS THERE because the pattern is SLATE, which can be written with these letters. This gives **57, a prime number**.

```
Total@LetterNumber@Characters["slate"]
```

```
57
```

36 – DIXIELAND (ch .16)

```
Total@LetterNumber@Characters["thought"]
```

```
99
```

37 – FOREST (ch .17)

```
Total@LetterNumber@Characters["marked"]
```

```
52
```

A **specific rule** applies. “The path is always clearly marked” refers to the text corresponding to the key **P A T H** in the original table. It contains 101 letters, a **prime number**.

```
StringLength[
 "SCARFAFORTUNECOOKIEANDATLASTHERTEASHETRIEDONTHESCARFSHESAIDANDGIVEITBACKTOHIM:
    THEOLDWAITERTOOKTHESCARF"]
```

```
101
```

38 – REHEARSAL (ch .18)

```
Total@LetterNumber@Characters["word"]
```

```
60
```

39 – PARTY (ch .19)

```
Total@LetterNumber@Characters["box"]
```

```
41
```

46 – HORSE (ch .21)

```
Total@LetterNumber@Characters["eclipsed"]
```

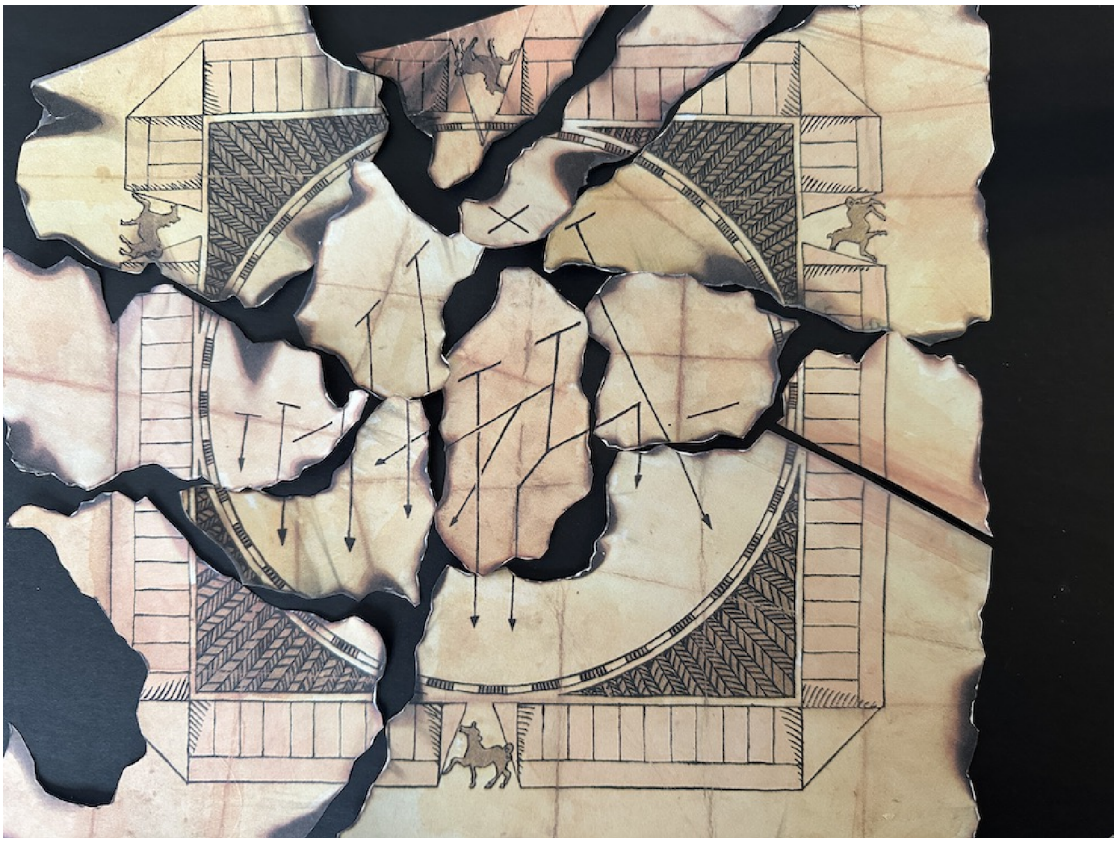
```
73
```

In conclusion, **the PRIME CHAPTERS are**:

- Chapter 1 - CREATION
- Chapter 3 - KITES
- Chapter 4 - CEMETERY
- Chapter 5 - FORTUNE
- Chapter 7 - ROAD
- Chapter 9 - MR. MAPS
- Chapter 15 - RIVER
- Chapter 17 - FOREST
- Chapter 19 - PARTY
- Chapter 21 - HORSE

The Map

The map is central to the solution. In the book, the map can be roughly reconstructed from 12 different pieces appearing on different pages.



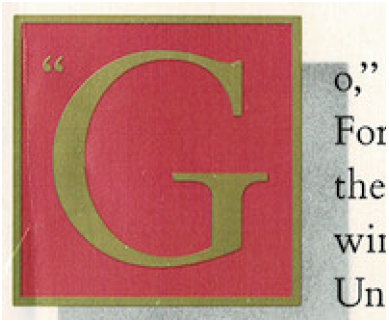
The fact that the map is in pieces and partly destroyed by fire indicates that all these arrows in the center are not precise vectors that would end up indicating some direction as the center circle is rotated. In fact, in one of the videos in the reference section (identified as Part 2), we are told that the arrangement of arrows corresponds to trees surrounding the golden horse location and that the X corresponds to this location.

For now, we will focus on the perimeter, where 4 horses appear to be running, all in the same direction. Somehow, this appears consistent with a “Journey”. There are 100 slots for letters around the map. Four of them are used by horses, and four more by the triangular corners. The map will reveal its other uses as we discover more clues.

The man who is also a bird

We stop shortly to consider the Man/Bird issue. He appears often, seemingly without reasons. He is often associated with music, sometimes guiding us, sometimes hiding things from us. What type of bird are we talking about here? Answering this question leads to surprising results affecting several aspects of the puzzle.

In REHEARSAL we have a man and a woman transforming into the Man with Black Gloves (TMWBG) and Amanda, while dancing to the music. Transformation is certainly one of the skills of TMWBG. At some point during this dance, there is a mention of "his partner in **pink**". Pink? Could this partner be the bird? There are very few pink birds, but a well known one is the **FLAMINGO**. Chapter 3 (KITES) starts with the word GO (page 16) on a bright pink background. On the other side of that page (page 17), aligned exactly with the pink square, is a kite showing a torch with a **FLAME**. This is the **FLAME IN GO**!



What is important here is that the flame is made up of four **sharp** points. And here is the link to music. A key signature with four sharps is E-major. And the four sharp notes of this key signature are **F#, G#, C#, D#**. The letters **C D F G** will play an important role more than once in the overall solution.

STARTING VERSION 2 MODIFICATIONS

The letters **C D F G** will play an important role more than once in the overall solution, particularly when music is heard.

Note that FLAMINGO (the bird) includes AMIGO (the man)

ENDING VERSION 2 MODIFICATIONS

The Journey

Chapter 5 - FORTUNE

FORTUNE is where we start our Journey.

```
(* The raw text of chapter 5 - FORTUNE *)

fortuneRaw =
  "On the other side of the mountains was a city by the sea. She had heard
    there was a house of fortune there. She had been told that the house
    was special, a place where you could ask strange questions and where
    you might be given answers. It was still light when Amanda arrived.
    Much to her surprise, she easily found the house of fortune. It was
    guarded by two stone lions. Above them a light shone in the window
    against the coming night, and the front door was opened to all seekers.
    The lions made her uneasy. Their stony snarls reminded her of the
    stones in the cemetery. But her own confusion, and the mystery of
    her missing Treasure, frightened her even more. She stepped across
    the threshold. At the end of a long hallway were narrow stairs that
    led to the room with the beckoning light. Amanda would come to learn
    that in each town where people lived there was always a house with
    an open door and a light in the window above. And though there were
    many such houses in many cities and towns, they all led to the same
    room, the room she now entered. Amanda was surprised by the nature of
    the light. Although night had fallen, the light in the room remained
    a little like dusk, a little like dawn. She couldn't tell which, or
    what would come next. Near the door she found a green felt table.
    On it lay instruments of luck and fate: a pair of black dice; three
    small, delicate bones; and three coins of ancient age. She picked
    up one of the coins. It was larger than the others, and worn by a
    million seeking hands. When she turned it over, she found pictures
    of twelve tiny animals around the rim. One of them was a horse.
    Against the wall stood an ancient fortune-telling machine. Inside its
    glass case sat a gypsy grandmother, made of wax, with lace-sleeved
    hands poised over curled cards. When Amanda approached, the machine
    whirled to life, and hidden gears guided mechanical hands in search
    of fate. But this is not what I came for, she thought. “And what did
    you come for,” asked a kind voice. Amanda turned and was surprised
    to find a woman with clear eyes and strong hands, sitting at a
    second table of green felt. Her dress was also lace-sleeved, but
    this woman was real.“I'm looking for my Treasure,” said Amanda, as
```

```
if it were obvious. “Ah,” replied the woman. “You can't expect to
find him right away. It's one step at a time. And you must collect
all the clues.” “Clues?” said Amanda. “How do I find clues?” “You
find them by taking The Journey.” Amanda was already tired from her
day's travel, and she certainly didn't feel like traveling anymore.
Reluctantly she asked, “How do I start The Journey?” “You must have
started already,” said the woman, “or you wouldn't be here.” “All I
want is to find my horse,” mumbled Amanda. She was sleepy now, and
she looked around for a place to rest her head. “Then beware the man
who is also a bird.” “What?” said Amanda, suddenly wide-awake. “Oh,
yes,” warned the woman. “He will try to lead you away from where
you want to go. He will lie with the truth, and make the true seem
false. Though the path is simple, it is well concealed.” Without
another word, the woman pulled a pack of cards from a green pouch
and dealt seven of them onto the table in front of Amanda. The first
card had a picture of a king dressed in royal cloth. He was seated
firmly on his throne, although from the dealer's point of view, he
was upside down. The second had a frightening skeleton. The third
had six empty cups and felt lonely indeed. The fourth showed a red
devil with a pitchfork. Next to him a woman grieved and hid her eyes.
That's no way to find clues, thought Amanda. The fifth card showed
a man serenading a woman beneath the moon, and Amanda felt better
about that. But in the sixth, a man fell past a turning waterwheel,
while a goddess looked the other way. The sun rose in the seventh
card and drenched the world in light. There was an eight card, with
a man on a horse. But the woman slipped it back into the deck before
Amanda could look at it. Seven cards were all a person was allowed.
“What do I do now?” the girl asked. “Each card is a door,” said
the woman. “I have dealt, but you must choose.” The girl did not
know which card to pick. She passed her hand from one to the next.
The light in the room seemed to change with each possible choice.
Finally Amanda chose the card with the wheel that turned and the
man that fell. The woman said not a word more. She merely smiled
her acceptance and closed her eyes. Whatever had happened, it was
over now. Amanda felt she should leave. Clutching her card to her
breast, she hurried from the room. The door immediately closed behind
her. In the room, the six cards remained on the table. Until a hand
in a black glove scooped them up and put them back into the deck.";
```

STARTING VERSION 2 MODIFICATIONS

Version 1 of FORTUNE has been extensively modified in Version 2, for various reasons. The most obvious was an error in the splitting of the text into sentences: sentence #27, identified as a single sentence in Version 1, actually included two sentences. The other was an attempt to streamline the whole

process of identifying states for The Journey. This resulted in the discovery of a new approach to select which sentences to consider in the chapter (the **selector**), thereby eliminating sentences that did not provide any useful data in Version 1. This approach lead to the understanding of several puzzles, including the infamous image in THE MAN WITH BLACK GLOVES and the text in LODGE.

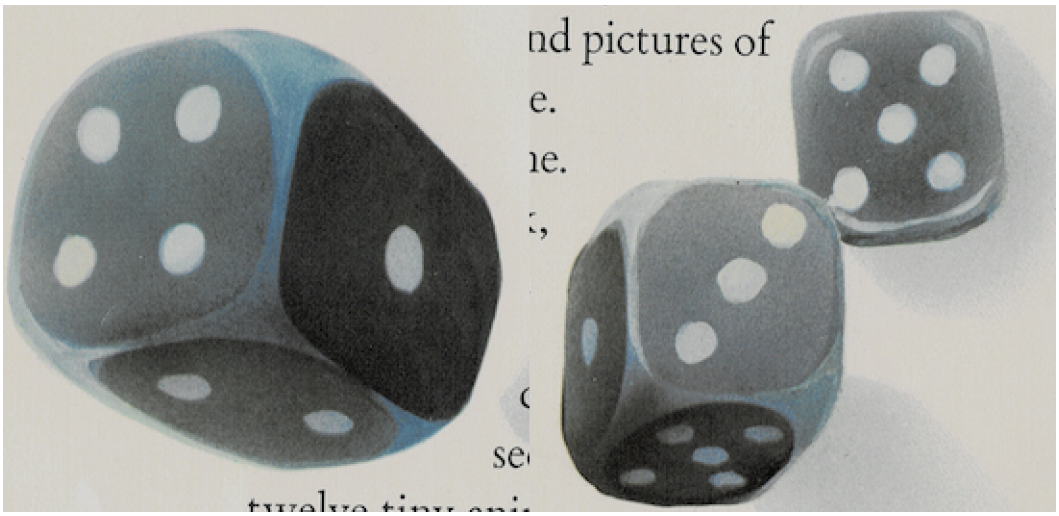
```
(* The full text of chapter FORTUNE split into numbered sentences *)

allSentences = Column@Normal@(assoc = AssociationThread[
       Range[Length[TextSentences[fortuneRaw]]], TextSentences[fortuneRaw]])
```

```
1 → On the other side of the mountains was a city by the sea.
2 → She had heard there was a house of fortune there.
3 → She had been told that the house was special, a place where you
    could ask strange questions and where you might be given answers.
4 → It was still light when Amanda arrived.
5 → Much to her surprise, she easily found the house of fortune.
6 → It was guarded by two stone lions.
7 → Above them a light shone in the window against the
    coming night, and the front door was opened to all seekers.
8 → The lions made her uneasy.
9 → Their stony snarls reminded her of the stones in the cemetery.
10 → But her own confusion, and the
    mystery of her missing Treasure, frightened her even more.
11 → She stepped across the threshold.
12 → At the end of a long hallway were
    narrow stairs that led to the room with the beckoning light.
13 → Amanda would come to learn that in each town where people lived there was
    always a house with an open door and a light in the window above.
14 → And though there were many such houses in many cities and
    towns, they all led to the same room, the room she now entered.
15 → Amanda was surprised by the nature of the light.
16 → Although night had fallen, the light in
    the room remained a little like dusk, a little like dawn.
17 → She couldn't tell which, or what would come next.
18 → Near the door she found a green felt table.
19 → On it lay instruments of luck and fate: a pair of black dice;
    three small, delicate bones; and three coins of ancient age.
20 → She picked up one of the coins.
21 → It was larger than the others, and worn by a million seeking hands.
22 → When she turned it over, she
    found pictures of twelve tiny animals around the rim.
23 → One of them was a horse.
24 → Against the wall stood an ancient fortune-telling machine.
```

```
25 → Inside its glass case sat a gypsy grandmother,
   made of wax, with lace-sleeved hands poised over curled cards.
26 → When Amanda approached, the machine whirled to life,
   and hidden gears guided mechanical hands in search of fate.
27 → But this is not what I came for, she thought.
28 → “And what did you come for,” asked a kind voice.
29 → Amanda turned and was surprised to find a woman with clear
   eyes and strong hands, sitting at a second table of green felt.
30 → Her dress was also lace-sleeved, but this woman was real.“
31 → I'm looking for my Treasure,” said Amanda, as if it were obvious.
32 → “Ah,” replied the woman.
33 → “You can't expect to find him right away.
34 → It's one step at a time.
35 → And you must collect all the clues.”
36 → “Clues?” said Amanda.
37 → “How do I find clues?”
38 → “You find them by taking The Journey.”
39 → Amanda was already tired from her day's
   travel, and she certainly didn't feel like traveling anymore.
40 → Reluctantly she asked, “How do I start The Journey?”
41 → “You must have started already,” said the woman, “or you wouldn't be here.”
42 → “All I want is to find my horse,” mumbled Amanda.
43 → She was sleepy now, and she looked around for a place to rest her head.
44 → “Then beware the man who is also a bird.”
45 → “What?” said Amanda, suddenly wide-awake.
46 → “Oh, yes,” warned the woman.
47 → “He will try to lead you away from where you want to go.
48 → He will lie with the truth, and make the true seem false.
49 → Though the path is simple, it is well concealed.”
50 →
 Without another word, the woman pulled a pack of cards from a green pouch and
   dealt seven of them onto the table in front of Amanda.
51 → The first card had a picture of a king dressed in royal cloth.
52 → He was seated firmly on his throne,
   although from the dealer's point of view, he was upside down.
53 → The second had a frightening skeleton.
54 → The third had six empty cups and felt lonely indeed.
55 → The fourth showed a red devil with a pitchfork.
56 → Next to him a woman grieved and hid her eyes.
57 → That's no way to find clues, thought Amanda.
58 → The fifth card showed a man serenading a
   woman beneath the moon, and Amanda felt better about that.
59 → But in the sixth, a man fell past a
   turning waterwheel, while a goddess looked the other way.
60 → The sun rose in the seventh card and drenched the world in light.
61 → There was an eight card, with a man on a horse.
62 → But the woman slipped it back into the deck before Amanda could look at it.
63 → Seven cards were all a person was allowed.
```

```
64 → “What do I do now?” the girl asked.
65 → “Each card is a door,” said the woman.
66 → “I have dealt, but you must choose.”
67 → The girl did not know which card to pick.
68 → She passed her hand from one to the next.
69 → The light in the room seemed to change with each possible choice.
70 →
 Finally Amanda chose the card with the wheel that turned and the man that fell.
71 → The woman said not a word more.
72 → She merely smiled her acceptance and closed her eyes.
73 → Whatever had happened, it was over now.
74 → Amanda felt she should leave.
75 → Clutching her card to her breast, she hurried from the room.
76 → The door immediately closed behind her.
77 → In the room, the six cards remained on the table.
78 →
 Until a hand in a black glove scooped them up and put them back into the deck.
```

The Journey is a trip (on paper) through the United States. How many “**spots”** will be visited during that trip? Just count the **spots** on the dice on page 25: **21 locations** will be visited.



What are these visited locations? Clues ares in these sentences:

- Sentence #3: a place to ask questions and may be get an answer
- Sentence #6: guarded by stone lions (an indication of higher rank in a society)
- Sentence #7: open to all seekers (a public place)
- Sentence #14: many such houses exist in many cities and town
- Sentence #13: each town where people lived there was always a house (the **people’s house**) with an open door and a light in the window above

It seems that we are talking about **state houses**. Therefore, our visited locations are **state capitals**.

The chapter contains 78 sentences, which is a very high number compared to most chapters. The sentence selector involves THE MAN WITH BLACK GLOVES. In sentences #44-48 we are warned about

him. In particular, sentence #48 tells us **He will lie with the truth, and make the true seem false**. As usual when having to look in other chapters to solve a prime chapter, images are important. The image in TMWBG is our starting point.

Here is the list of things to consider in this image:

- The focus is on 3 “objects”, from left to right: the monolith, an opening in the wall and a mirror.
- Viewed as a group, the above three objects look like a 100.
- The huge glove reminds us of an elephant **trunk** and the clouds on the monolith are shaped like an elephant head and **trunk** (the elephant has a bird in its mouth).
- We see trees in the wall opening, with emphasis on a huge tree **trunk**.
- On the far left, we see columns and a horse.
- The floor reminds us of rows in a table.

Clearly, **trunk** is a reference to **truncate**.
The proper interpretation of this image rests on understanding why the monolith from **2001: A Space Odyssey** floats above the ground. The reason is that 2001 is a composite number. But we need to represent this number with **monolithic numbers**, which is another way of saying **prime factors**. We are in effect truncating a big number into three shorter ones.

3 *x* 23 *x* 29 = 2001

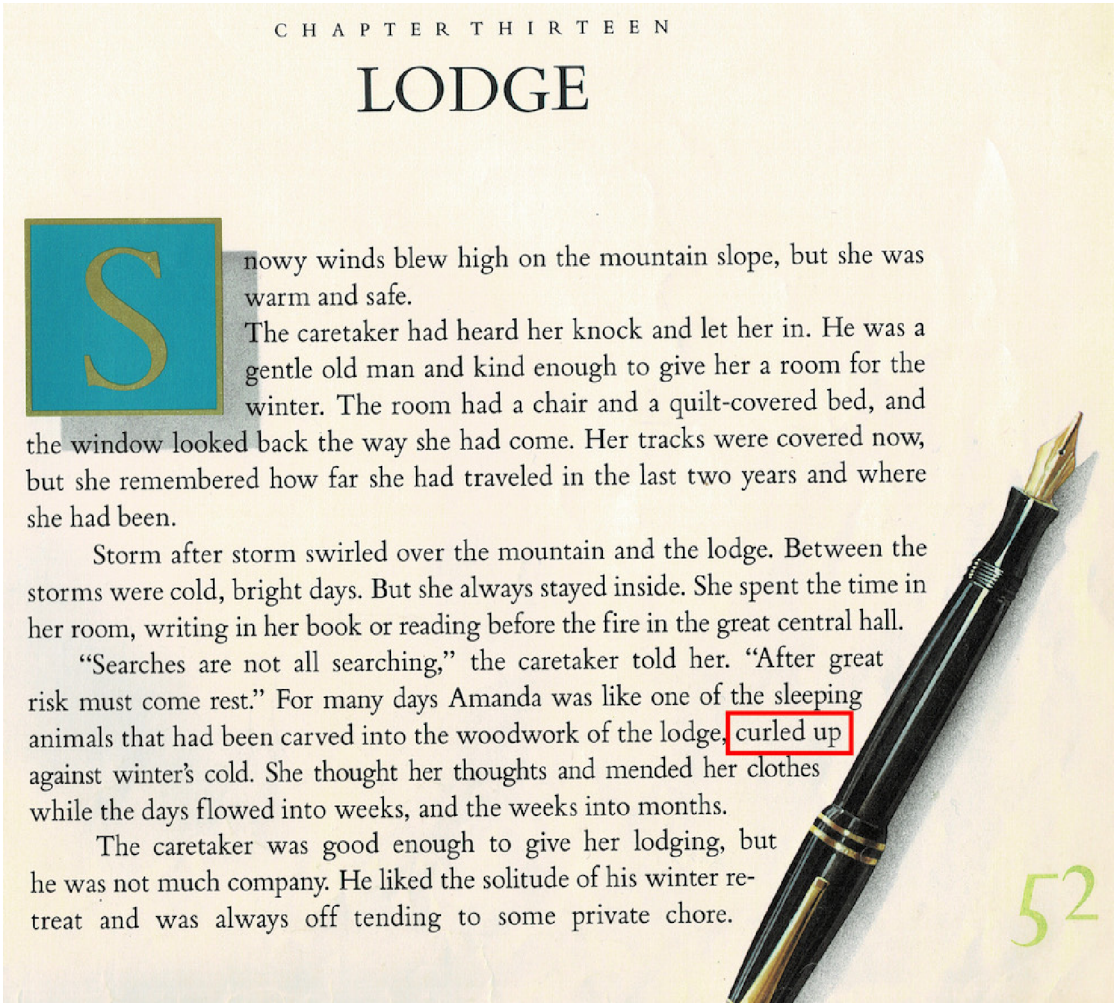
We move to the center opening in the wall with the numbers 3, 23, 29. I believe the tree trunk is to be read as "**three** trunk" , so we should **truncate number 3**. This leaves us with:

23 *x* 29 = 667

Like in COURTYARD, we find ourselves again with the multiplication of two prime numbers. And again, there is the possibility of a table, a 23 x 29 table. The third object is the rectangular mirror, which would indicate that rows (23) are smaller than columns (29). But where do the 667 characters come from?

The answer is in the mirror. Where do we find a bedroom with Amanda, a window and a horse in a wooden frame like the mirror? In LODGE. Page 53.

Here we have Amanda in a bedroom, writing, and a wooden frame with an upside down state of North Carolina being taken out. This is a clue that whatever table we put together, we have to focus on the frame. While writing, Amanda is looking at the window, which looks like a table.
What about the 667 characters supposedly making up this table? The mirror in TMWBG is a flipping mirror, so **we must flip the page to page 52**.



CHAPTER THIRTEEN

LODGE

Snowy winds blew high on the mountain slope, but she was warm and safe.

The caretaker had heard her knock and let her in. He was a gentle old man and kind enough to give her a room for the winter. The room had a chair and a quilt-covered bed, and the window looked back the way she had come. Her tracks were covered now, but she remembered how far she had traveled in the last two years and where she had been.

Storm after storm swirled over the mountain and the lodge. Between the storms were cold, bright days. But she always stayed inside. She spent the time in her room, writing in her book or reading before the fire in the great central hall.

"Searches are not all searching," the caretaker told her. "After great risk must come rest." For many days Amanda was like one of the sleeping animals that had been carved into the woodwork of the lodge, curled up against winter's cold. She thought her thoughts and mended her clothes while the days flowed into weeks, and the weeks into months.

The caretaker was good enough to give her lodging, but he was not much company. He liked the solitude of his winter retreat and was always off tending to some private chore.

52

This page has a pen forming a triangle with the edge of the page, similar to what we saw before in COURTYARD. And although the page is not "curled up", the actual words "curled up" appear right there in the text. This seems to be a clue that the text of the whole page will be used.

```
(* raw text of page 52 *)
page52Raw =
  "Snowy winds blew high on the mountain slope, but she was warm and safe. The
    caretaker had heard her knock and let her in. He was a gentle old
    man and kind enough to give her a room for the winter. The room had
    a chair and a quilt-covered bed, and the window looked back the way
    she had come. Her tracks were covered now, but she remembered how
    far she had traveled in the last two years and where she had been.
    Storm after storm swirled over the mountain and the lodge. Between
    the storms were cold, bright days. But she always stayed inside. She
    spent the time in her room, writing in her book or reading before the
    fire in the great central hall. “Searches are not all searching,”
    the caretaker told her. “After great risk must come rest.” For many
    days Amanda was like one of the sleeping animals that had been carved
    into the woodwork of the lodge, curled up against winter's cold. She
    thought her thoughts and mended her clothes while the days flowed into
    weeks, and the weeks into months. The caretaker was good enough to
    give her lodging, but he was not much company. He liked the solitude
    of his winter retreat and was always off tending some private chore.";
```

```
Length@StringCases[page52Raw, LetterCharacter]
```

```
919
```

```
Column@Normal@(ass = AssociationThread[
      Range[Length[TextSentences[page52Raw]]], TextSentences[page52Raw]])
```

```
1 → Snowy winds blew high on the mountain slope, but she was warm and safe.
2 → The caretaker had heard her knock and let her in.
3 → He was a gentle old man and kind enough to give her a room for the winter.
4 → The room had a chair and a quilt-covered
    bed, and the window looked back the way she had come.
5 → Her tracks were covered now, but she remembered how far
    she had traveled in the last two years and where she had been.
6 → Storm after storm swirled over the mountain and the lodge.
7 → Between the storms were cold, bright days.
8 → But she always stayed inside.
9 → She spent the time in her room, writing in her
    book or reading before the fire in the great central hall.
10 → “Searches are not all searching,” the caretaker told her.
11 → “After great risk must come rest.”
12 →
 For many days Amanda was like one of the sleeping animals that had been carved
    into the woodwork of the lodge, curled up against winter's cold.
13 → She thought her thoughts and mended her clothes
    while the days flowed into weeks, and the weeks into months.
14 →
 The caretaker was good enough to give her lodging, but he was not much company.
15 → He liked the solitude of his winter
    retreat and was always off tending some private chore.
```

We have 919 letters in 15 sentences, much more than the 667 letters that we need to build a table. But there are clues that 3 sentences should be eliminated:

- Sentence #12: Amanda is sleeping and animals are carved out
- Sentence #14: the caretaker is not much company (so he does not want to be with other sentences)
- Sentence #15: he liked solitude and is off tending some private core.

We remove these 3 sentences.

```
Column@(rp = ReplacePart[Normal@ass, ({#} & /@ {12, 14, 15}) → Nothing])
```

```
1 → Snowy winds blew high on the mountain slope, but she was warm and safe.
2 → The caretaker had heard her knock and let her in.
3 → He was a gentle old man and kind enough to give her a room for the winter.
4 → The room had a chair and a quilt-covered
    bed, and the window looked back the way she had come.
5 → Her tracks were covered now, but she remembered how far
    she had traveled in the last two years and where she had been.
6 → Storm after storm swirled over the mountain and the lodge.
7 → Between the storms were cold, bright days.
8 → But she always stayed inside.
9 → She spent the time in her room, writing in her
    book or reading before the fire in the great central hall.
10 → “Searches are not all searching,” the caretaker told her.
11 → “After great risk must come rest.”
13 → She thought her thoughts and mended her clothes
    while the days flowed into weeks, and the weeks into months.
```

```
sj = StringJoin@Values@rp;
Length@StringCases[sj, LetterCharacter]
```

```
667
```

We have our 667 letters. We can build our table.

```
Grid[Partition[ToUpperCase@StringCases[sj, LetterCharacter], 23], Frame → All,
 ItemSize → 1, Background → {{1 → Cyan, 23 → Cyan}, {1 → Cyan, 29 → Cyan}}]
```

S	N	O	W	Y	W	I	N	D	S	B	L	E	W	H	I	G	H	O	N	T	H	E
M	O	U	N	T	A	I	N	S	L	O	P	E	B	U	T	S	H	E	W	A	S	W
A	R	M	A	N	D	S	A	F	E	T	H	E	C	A	R	E	T	A	K	E	R	H
A	D	H	E	A	R	D	H	E	R	K	N	O	C	K	A	N	D	L	E	T	H	E
R	I	N	H	E	W	A	S	A	G	E	N	T	L	E	O	L	D	M	A	N	A	N
D	K	I	N	D	E	N	O	U	G	H	T	O	G	I	V	E	H	E	R	A	R	O
O	M	F	O	R	T	H	E	W	I	N	T	E	R	T	H	E	R	O	O	M	H	A
D	A	C	H	A	I	R	A	N	D	A	Q	U	I	L	T	C	O	V	E	R	E	D
B	E	D	A	N	D	T	H	E	W	I	N	D	O	W	L	O	O	K	E	D	B	A
C	K	T	H	E	W	A	Y	S	H	E	H	A	D	C	O	M	E	H	E	R	T	R
A	C	K	S	W	E	R	E	C	O	V	E	R	E	D	N	O	W	B	U	T	S	H
E	R	E	M	E	M	B	E	R	E	D	H	O	W	F	A	R	S	H	E	H	A	D
T	R	A	V	E	L	E	D	I	N	T	H	E	L	A	S	T	T	W	O	Y	E	A
R	S	A	N	D	W	H	E	R	E	S	H	E	H	A	D	B	E	E	N	S	T	O
R	M	A	F	T	E	R	S	T	O	R	M	S	W	I	R	L	E	D	O	V	E	R
T	H	E	M	O	U	N	T	A	I	N	A	N	D	T	H	E	L	O	D	G	E	B
E	T	W	E	E	N	T	H	E	S	T	O	R	M	S	W	E	R	E	C	O	L	D
B	R	I	G	H	T	D	A	Y	S	B	U	T	S	H	E	A	L	W	A	Y	S	S
T	A	Y	E	D	I	N	S	I	D	E	S	H	E	S	P	E	N	T	T	H	E	T
I	M	E	I	N	H	E	R	R	O	O	M	W	R	I	T	I	N	G	I	N	H	E
R	B	O	O	K	O	R	R	E	A	D	I	N	G	B	E	F	O	R	E	T	H	E
F	I	R	E	I	N	T	H	E	G	R	E	A	T	C	E	N	T	R	A	L	H	A
L	L	S	E	A	R	C	H	E	S	A	R	E	N	O	T	A	L	L	S	E	A	R
C	H	I	N	G	T	H	E	C	A	R	E	T	A	K	E	R	T	O	L	D	H	E
R	A	F	T	E	R	G	R	E	A	T	R	I	S	K	M	U	S	T	C	O	M	E
R	E	S	T	S	H	E	T	H	O	U	G	H	T	H	E	R	T	H	O	U	G	H
T	S	A	N	D	M	E	N	D	E	D	H	E	R	C	L	O	T	H	E	S	W	H
I	L	E	T	H	E	D	A	Y	S	F	L	O	W	E	D	I	N	T	O	W	E	E
K	S	A	N	D	T	H	E	W	E	E	K	S	I	N	T	O	M	O	N	T	H	S

The border contains the important letters, as emphasized by the wooden frames of both the mirror in TMWBG and the image in LODGE. **It contains 100 letters**, matching the number 100 observed in TMWBG.
(Yes, I know, it is tempting to use these letters to fill the map border, but we will not do that).
We will collect the 100-letter string around the map, starting with “Snowy winds...”

```
frameString =
  "SNOWYWINDSBLEWHIGHONTHEWHENOADARHDAORBDSTEEAREEHHESHTNOMOTNISKEEWEHTDNASKITR\
     RCLFRITBETRRTEACBDODRAAM";
```

Each of the letters is then converted to a number, using its position in the alphabet. This is our **selector**.

```
frameNumbers = LetterNumber@StringCases[frameString, LetterCharacter]
```

```
{19, 14, 15, 23, 25, 23, 9, 14, 4, 19, 2, 12, 5, 23, 8, 9, 7, 8, 15, 14, 20, 8, 5, 23, 8,
 5, 14, 15, 1, 4, 1, 18, 8, 4, 1, 15, 18, 2, 4, 19, 20, 5, 5, 1, 18, 5, 5, 8, 8, 5, 19,
 8, 20, 14, 15, 13, 15, 20, 14, 9, 19, 11, 5, 5, 23, 5, 8, 20, 4, 14, 1, 19, 11, 9, 20,
 18, 18, 3, 12, 6, 18, 9, 20, 2, 5, 20, 18, 18, 20, 5, 1, 3, 2, 4, 15, 4, 18, 1, 1, 13}
```

We then use these 100 numbers to count from the beginning of our 78 sentences in FORTUNE. This requires going over the chapter's sentences several times. We end up with sentence numbers for sentences selected by this process. Note that we remove duplicates. These sentences will be used in the next step.

```
range = Range[78];
selectedForProcessing = DeleteDuplicates[(range = RotateLeft[range, #];
      v = range[[1]] - 1;
      If[v == 0, v = 78];
      v) & /@ frameNumbers]
```

```
{19, 33, 48, 71, 18, 41, 50, 64, 68, 9, 11, 23, 28, 51, 59, 75, 5, 20, 34, 54,
 62, 67, 12, 25, 39, 55, 60, 78, 8, 13, 46, 52, 24, 42, 47, 73, 14, 22, 56, 6,
 21, 16, 26, 49, 4, 53, 40, 43, 61, 1, 10, 30, 32, 37, 57, 15, 35, 44, 65, 69}
```

Here is the sorted list of sentence numbers.

```
Sort[selectedForProcessing]
```

```
{1, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26,
 28, 30, 32, 33, 34, 35, 37, 39, 40, 41, 42, 43, 44, 46, 47, 48, 49, 50, 51,
 52, 53, 54, 55, 56, 57, 59, 60, 61, 62, 64, 65, 67, 68, 69, 71, 73, 75, 78}
```

We have selected a total of 60 sentences.

```
Length[selectedForProcessing]
```

```
60
```

Here is the final list of sentences, with sentence number.

```
finalSentencesKept = Column@((Normal@assoc)[[#]] & /@ Sort[selectedForProcessing])
```

```
1 → On the other side of the mountains was a city by the sea.
4 → It was still light when Amanda arrived.
5 → Much to her surprise, she easily found the house of fortune.
6 → It was guarded by two stone lions.
```

```
8 → The lions made her uneasy.
9 → Their stony snarls reminded her of the stones in the cemetery.
10 → But her own confusion, and the
   mystery of her missing Treasure, frightened her even more.
11 → She stepped across the threshold.
12 → At the end of a long hallway were
   narrow stairs that led to the room with the beckoning light.
13 → Amanda would come to learn that in each town where people lived there was
   always a house with an open door and a light in the window above.
14 → And though there were many such houses in many cities and
   towns, they all led to the same room, the room she now entered.
15 → Amanda was surprised by the nature of the light.
16 → Although night had fallen, the light in
   the room remained a little like dusk, a little like dawn.
18 → Near the door she found a green felt table.
19 → On it lay instruments of luck and fate: a pair of black dice;
   three small, delicate bones; and three coins of ancient age.
20 → She picked up one of the coins.
21 → It was larger than the others, and worn by a million seeking hands.
22 → When she turned it over, she
   found pictures of twelve tiny animals around the rim.
23 → One of them was a horse.
24 → Against the wall stood an ancient fortune-telling machine.
25 → Inside its glass case sat a gypsy grandmother,
   made of wax, with lace-sleeved hands poised over curled cards.
26 → When Amanda approached, the machine whirled to life,
   and hidden gears guided mechanical hands in search of fate.
28 → “And what did you come for,” asked a kind voice.
30 → Her dress was also lace-sleeved, but this woman was real.“
32 → “Ah,” replied the woman.
33 → “You can't expect to find him right away.
34 → It's one step at a time.
35 → And you must collect all the clues.”
37 → “How do I find clues?”
39 → Amanda was already tired from her day's
   travel, and she certainly didn't feel like traveling anymore.
40 → Reluctantly she asked, “How do I start The Journey?”
41 → “You must have started already,” said the woman, “or you wouldn't be here.”
42 → “All I want is to find my horse,” mumbled Amanda.
43 → She was sleepy now, and she looked around for a place to rest her head.
44 → “Then beware the man who is also a bird.”
46 → “Oh, yes,” warned the woman.
47 → “He will try to lead you away from where you want to go.
48 → He will lie with the truth, and make the true seem false.
49 → Though the path is simple, it is well concealed.”
```

```
50 →
 Without another word, the woman pulled a pack of cards from a green pouch and
   dealt seven of them onto the table in front of Amanda.
51 → The first card had a picture of a king dressed in royal cloth.
52 → He was seated firmly on his throne,
   although from the dealer's point of view, he was upside down.
53 → The second had a frightening skeleton.
54 → The third had six empty cups and felt lonely indeed.
55 → The fourth showed a red devil with a pitchfork.
56 → Next to him a woman grieved and hid her eyes.
57 → That's no way to find clues, thought Amanda.
59 → But in the sixth, a man fell past a
   turning waterwheel, while a goddess looked the other way.
60 → The sun rose in the seventh card and drenched the world in light.
61 → There was an eight card, with a man on a horse.
62 → But the woman slipped it back into the deck before Amanda could look at it.
64 → “What do I do now?” the girl asked.
65 → “Each card is a door,” said the woman.
67 → The girl did not know which card to pick.
68 → She passed her hand from one to the next.
69 → The light in the room seemed to change with each possible choice.
71 → The woman said not a word more.
73 → Whatever had happened, it was over now.
75 → Clutching her card to her breast, she hurried from the room.
78 →
 Until a hand in a black glove scooped them up and put them back into the deck.
```

We finally arrive at the process involved in determining visited locations during the Journey.

Sentence #1 specifies that Amanda arrived in “a city by the sea”. This is a play on words indicating **Atlantic City**. This also matches the theme of a casino. Atlantic City is a city in New Jersey, but it is not the state capital. The state capital is Trenton.

This information tells us to start our Journey in New Jersey. Sentence #40 contains the text **“start The Journey”**. This is where our Journey starts.
The word Journey contains the 2 letters **NJ**, the state abbreviation for New Jersey. Note that the letter J has a very low frequency in English. The main reason to bring up a “Journey” is likely to provide this NJ abbreviation.

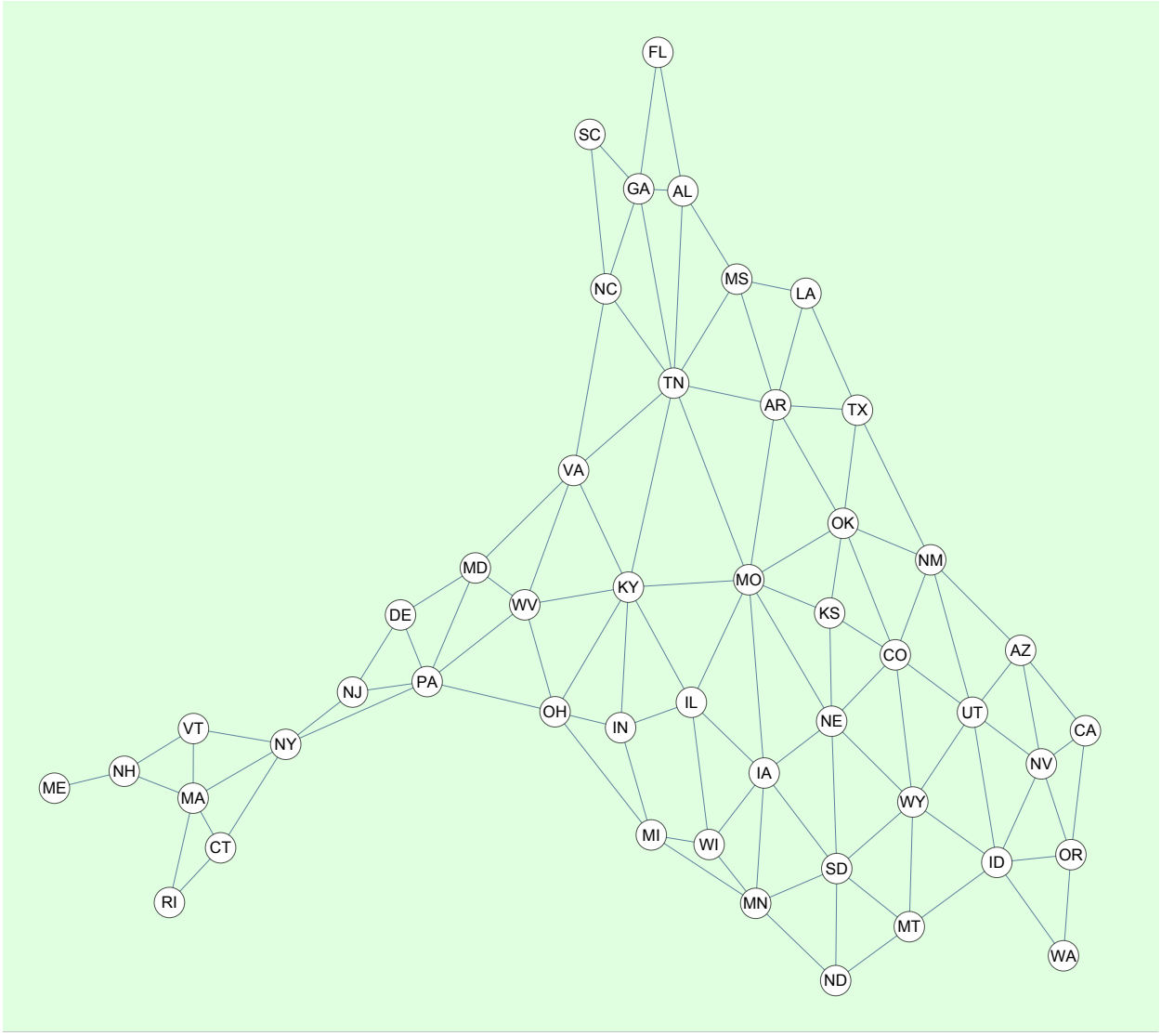
Here is the method used to collect places visited in the Journey:

- Sentences are processed in order, starting with sentence #40.
- In each sentence, we need to scan each word, looking for letters making up the abbreviation of states.

- If a word contains letters corresponding to a 2-letter state abbreviation, **and this abbreviation identifies a state contiguous with our current state in The Journey**, we add it as our next destination, **unless specific rules are dictated by the text of the sentence being examined**. A specific rules may be a riddle of some sort telling us how to proceed to identify the correct state, or sometimes a way to tell us that the sentence should be ignored.
- A state cannot be visited more than once. After that, its abbreviation is ignored
- As we will see, this process is assisted by *Mathematica*. The program identifies applicable state abbreviations found in each words of a sentence, based on our current location (i.e. state) in the Journey and where we have been before (states already visited).
  - **WARNING**: some sentences contain more than one correct state (i.e. they contain a string of contiguous states). Such strings may be missed by the software. For example, if we are in state A, then state B (contiguous to A) may be identified, but the sentence may also contain state C (contiguous to B, but not to A). In this case, state C is valid but will not be automatically identified by the software.

To use the software assisted approach, we need to compile basic informations about state abbreviations and the contiguity of states. The contiguity is represented below as a graph.

```
(* List of states, state contiguity, and graph showing states contiguity *)

stateList = {"AL", "FL", "GA", "MS", "TN", "AR", "LA", "MO", "OK", "TX", "AZ", "CA",
    "NM", "NV", "UT", "OR", "CO", "KS", "NE", "WY", "CT", "MA", "NY", "RI",
    "MD", "VA", "DE", "NJ", "PA", "NC", "SC", "IA", "IL", "MN", "SD", "WI",
    "ID", "MT", "WA", "IN", "KY", "MI", "OH", "WV", "NH", "VT", "ME", "ND"};

stateContiguity = {"AL" ⟷ "FL", "AL" ⟷ "GA", "AL" ⟷ "MS", "AL" ⟷ "TN", "AR" ⟷ "LA",
    "AR" ⟷ "MO", "AR" ⟷ "MS", "AR" ⟷ "OK", "AR" ⟷ "TN", "AR" ⟷ "TX", "AZ" ⟷ "CA",
    "AZ" ⟷ "NM", "AZ" ⟷ "NV", "AZ" ⟷ "UT", "CA" ⟷ "NV", "CA" ⟷ "OR", "CO" ⟷ "KS",
    "CO" ⟷ "NE", "CO" ⟷ "NM", "CO" ⟷ "OK", "CO" ⟷ "UT", "CO" ⟷ "WY", "CT" ⟷ "MA",
    "CT" ⟷ "NY", "CT" ⟷ "RI", "DE" ⟷ "MD", "DE" ⟷ "NJ", "DE" ⟷ "PA", "FL" ⟷ "GA",
    "GA" ⟷ "NC", "GA" ⟷ "SC", "GA" ⟷ "TN", "IA" ⟷ "IL", "IA" ⟷ "MN", "IA" ⟷ "MO",
    "IA" ⟷ "NE", "IA" ⟷ "SD", "IA" ⟷ "WI", "ID" ⟷ "MT", "ID" ⟷ "NV", "ID" ⟷ "OR",
    "ID" ⟷ "UT", "ID" ⟷ "WA", "ID" ⟷ "WY", "IL" ⟷ "IN", "IL" ⟷ "KY", "IL" ⟷ "MO",
    "IL" ⟷ "WI", "IN" ⟷ "KY", "IN" ⟷ "MI", "IN" ⟷ "OH", "KS" ⟷ "MO", "KS" ⟷ "NE",
    "KS" ⟷ "OK", "KY" ⟷ "MO", "KY" ⟷ "OH", "KY" ⟷ "TN", "KY" ⟷ "VA", "KY" ⟷ "WV",
    "LA" ⟷ "MS", "LA" ⟷ "TX", "MA" ⟷ "NH", "MA" ⟷ "NY", "MA" ⟷ "RI", "MA" ⟷ "VT",
    "MD" ⟷ "PA", "MD" ⟷ "VA", "MD" ⟷ "WV", "ME" ⟷ "NH", "MI" ⟷ "OH", "MI" ⟷ "WI",
    "MI" ⟷ "MN", "MN" ⟷ "ND", "MN" ⟷ "SD", "MN" ⟷ "WI", "MO" ⟷ "NE", "MO" ⟷ "OK",
    "MO" ⟷ "TN", "MS" ⟷ "TN", "MT" ⟷ "ND", "MT" ⟷ "SD", "MT" ⟷ "WY", "NC" ⟷ "SC",
    "NC" ⟷ "TN", "NC" ⟷ "VA", "ND" ⟷ "SD", "NE" ⟷ "SD", "NE" ⟷ "WY", "NH" ⟷ "VT",
    "NJ" ⟷ "NY", "NJ" ⟷ "PA", "NM" ⟷ "OK", "NM" ⟷ "TX", "NM" ⟷ "UT", "NV" ⟷ "OR",
    "NV" ⟷ "UT", "NY" ⟷ "PA", "NY" ⟷ "VT", "OH" ⟷ "PA", "OH" ⟷ "WV", "OK" ⟷ "TX",
    "OR" ⟷ "WA", "PA" ⟷ "WV", "SD" ⟷ "WY", "TN" ⟷ "VA", "UT" ⟷ "WY", "VA" ⟷ "WV"};

stateGraph =
 Graph[stateContiguity, VertexLabels → Placed[Automatic, Center], VertexSize → .5,
  ImageSize → 620, AspectRatio → 0.9, VertexShapeFunction → ({White,
      EdgeForm[Black], Rectangle[# - .2, # + .2, RoundingRadius → .2], Black} &)]
```



The above graph represents the connections between states. Mathematica will use it, but it is also useful for manual processing, if desired. Even when using Mathematica, it may be useful to refer to this graph to understand specific rules or resolve issues such as the one mentioned in the above WARNING.

We will now create two *Mathematica* functions that will help us process the sentences, one at a time. We cannot blindly run them all in a batch as there are specific rules, where interpretation as to what to do is linked to the clues present in the sentences themselves. Taken together, these functions ensure that only possible states (contiguous and not visited before) are considered at every step along the way.

```
getPossibleStates[lastState_, noStates_] := Module[{vlist},
  vlist = VertexList[NeighborhoodGraph[stateGraph, lastState]];
  Complement[vlist, noStates]
 ]


processStateSentences[sentenceNumber_, possibleStates_, rawText_] :=
 Module[{sentence, tup, chs, inter, interJ},
  sentence = ToUpperCase[(TextSentences@rawText)⟦sentenceNumber⟧];
  tup = Tuples[Characters@#, {2}] & /@ TextWords@sentence;
  chs = Characters[#] & /@ stateList;
  inter = Partition[
    Flatten@(If[Length@Intersection[(Characters[#] & /@ stateList), #] > 0,
          Intersection[(Characters[#] & /@ stateList), #], Nothing] & /@ tup), 2];
  interJ = StringJoin[#] & /@ inter;
  Intersection[interJ, possibleStates]
 ]
```

These functions will be called for each sentence being processed. It may be confusing to understand what is going on with these functions, so we will explain in more details:

- **processStateSentences** is the main function.
  - The function takes 3 arguments
    - the **raw text** of the chapter. This includes all the sentences.
    - the **sentence number** of the sentence being processed. This number will come from our selected 60 sentences, starting with sentence #40
    - the list of **possible states** to look for in the sentence. There are two ways of providing this information:
    - 1) a list of state abbreviations
    - 2) calling the function **getPosssibleStates** to compute that list of possible state abbreviations
  - The function does the following
    - Finds the sentence text from the **raw text** and the **sentence number**
    - Process words of the sentence to Identify any state abbreviations present
    - Keep only the abbreviations that are in the list of **possible states**
  - The function returns a list of applicable abbreviations.

- **getPosssibleStates** is the secondary function.
  - The function takes 2 arguments

- the **last state** visited. This is our current location in the Journey.
- the **no state** list, which is a list of state abbreviations **not** to be considered. This is where we will list states already visited.

- The function does the following:
  - From the contiguity graph, it finds the list of states contiguous to the **last state** (i.e. our current location)
  - The **no state** list is then removed from that list of contiguous states.
- The function returns a list of **possible states**, to be used as the second argument in **processStateSentences**

We will now process each sentence by calling **processStateSentences**, starting with sentence #40. Note that for this particular sentence we use the list {“NJ”} as the second argument, as we know that the starting point is New Jersey. As long as NJ is present in a word of the sentence (i.e. Journey) the function will return it as the only possible answer. For other sentences, we use a call to **getPosssibleStates** as the second argument to compute the possible states.

40

```
processStateSentences[40, {"NJ"}, fortuneRaw]
```

```
{NJ}
```

Current Journey: NJ

41

```
processStateSentences[41, getPossibleStates["NJ", {"NJ"}], fortuneRaw]
```

```
{DE}
```

Current Journey: NJ-DE

42

```
processStateSentences[42, getPossibleStates["DE", {"NJ", "DE"}], fortuneRaw]
```

```
{MD}
```

Current Journey: NJ-DE-MD

43

```
processStateSentences[43, getPossibleStates["MD", {"NJ", "DE", "MD"}], fortuneRaw]
```

```
{PA}
```

Sentence #43 tells us that “she looked around for a place to rest her **head**”. This is a way to indicate a **heading**. Of the 3 possible states (PA, VA, WV), only one includes a heading direction: **West Virginia (WV).**
What about PA, obtained from the word “place”? The sentence starts with “She was **sleepy** now”. This is to be understood as **sleep P**. So the P is not to be considered, thereby eliminating PA.
We select **WV**.

Current Journey: NJ-DE-MD-WV

44

```
processStateSentences[44,
 getPossibleStates["WV", {"NJ", "DE", "MD", "WV"}], fortuneRaw]
```

```
{OH}
```

This sentence is a warning to be careful (“Beware”). So we will ignore this result, even if OH is the right answer, as we will see in the next sentence.

Current Journey: NJ-DE-MD-WV

46

```
processStateSentences[46,
 getPossibleStates["WV", {"NJ", "DE", "MD", "WV"}], fortuneRaw]
```

```
{OH}
```

Current Journey: NJ-DE-MD-WV-OH

47 – 48

The text of sentences 48-49 clearly indicates that these should not be considered.

Current Journey: NJ-DE-MD-WV-OH

49

```
processStateSentences[49,
 getPossibleStates["OH", {"NJ", "DE", "MD", "WV", "OH"}], fortuneRaw]
```

```
{MI, PA}
```

Recalling from sentence 43 that P is asleep (sleep P), the correct state is MI.

Current Journey: NJ-DE-MD-WV-OH-MI

50

```
processStateSentences[50,
 getPossibleStates["MI", {"NJ", "DE", "MD", "WV", "OH", "MI"}], fortuneRaw]
```

```
{IN, MN, WI}
```

The correct answer is **MN-ND**, both from the word “Amanda”. **ND** is not picked up by Mathematica because it is contiguous with **MN**, not **MI**. We get two states from this sentence.

The **WI** comes from “**Without** a word”, indicating that that we have to proceed “without” this word. The rest of the sentence tells us that seven cards were pulled... in front of Amanda. Removal of the 7 words before the word Amanda removes **IN**.

This is a clever sentence. We finished sentence #49 with **MI**, and we are now jumping to **MN** (instead of the obvious **WI**). **MI** and **MN** are not contiguous on land, but share a common border in Lake Superior.

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND

51

```
processStateSentences[51, getPossibleStates["ND",
  {"NJ", "DE", "MD", "WV", "OH", "MI", "MN", "ND"}], fortuneRaw]
```

```
{SD}
```

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD

52

```
processStateSentences[52, getPossibleStates["SD",
  {"NJ", "DE", "MD", "WV", "OH", "MI", "MN", "ND", "SD"}], fortuneRaw]
```

```
{NE}
```

In sentence #52, we get **NE** from the word “throne”, but we need a different “point of view”. The rest of the sentence ends with “he was **upside down**”. If we start in Nebraska (NE) and go **UP-SIDE-DOWN** on a map, we move **up** to **SD** - move **side**ways to **MN** - move down to **IOWA (IA)**.
IA is the state we are looking for. (I don’t have an immediate explanation as to why the side move is to the EAST rather than the WEST).

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA

53

```
processStateSentences[53, getPossibleStates["IA",
  {"NJ", "DE", "MD", "WV", "OH", "MI", "MN", "ND", "SD", "IA"}], fortuneRaw]
```

```
{NE}
```

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE

54 – 55

The text of sentences 54-55 clearly indicates that these should not be considered. Sentence #54 is “empty” and sentence #55 is the devil.

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE

56

```
processStateSentences[56, getPossibleStates["NE",
  {"NJ", "DE", "MD", "WV", "OH", "MI", "MN", "ND", "SD", "IA", "NE"}], fortuneRaw]
```

```
{MO}
```

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE-MO

57

Sentence #57 plainly tells us that there is no clue there.

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE-MO

59

```
processStateSentences[59, getPossibleStates["MO", {"NJ", "DE", "MD",
   "WV", "OH", "MI", "MN", "ND", "SD", "IA", "NE", "MO"}], fortuneRaw]
```

```
{AR, IL, OK, TN}
```

Sentence #59 tells us that “a man fell past a turning waterwheel”. This means that the rest of the sentence (past waterwheel) falls. This removes AR, IL and OK.

The sentence removed is about “looking the other way”.
We select **TN**.

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE-MO-TN

60

```
processStateSentences[60,
 getPossibleStates["TN", {"NJ", "DE", "MD", "WV", "OH", "MI",
   "MN", "ND", "SD", "IA", "NE", "MO", "TN"}], fortuneRaw]
```

```
{AR, NC}
```

The first state to appear is **AR** is “card”.
Then appears **NC** (not contiguous with AR, so cannot be chosen if AR is chosen)) in the word “drenched”, which has the sense of “completely **covered** ”. (We recall that North Carolina was also removed from the frame in LODGE).
We select **AR**.

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE-MO-TN-AR

61

This is the “eighth card”. We are told in the next sentence (**#62**) not to look at it.

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE-MO-TN-AR

62

```
processStateSentences[62,
 getPossibleStates["AR", {"NJ", "DE", "MD", "WV", "OH", "MI",
   "MN", "ND", "SD", "IA", "NE", "MO", "TN", "AR"}], fortuneRaw]
```

```
{OK}
```

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE-MO-TN-AR-OK

64

```
processStateSentences[64,
 getPossibleStates["OK", {"NJ", "DE", "MD", "WV", "OH", "MI",
   "MN", "ND", "SD", "IA", "NE", "MO", "TN", "AR", "OK"}], fortuneRaw]
```

```
{KS}
```

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE-MO-TN-AR-OK-KS

65

```
processStateSentences[65,
 getPossibleStates["KS", {"NJ", "DE", "MD", "WV", "OH", "MI", "MN",
   "ND", "SD", "IA", "NE", "MO", "TN", "AR", "OK", "KS"}], fortuneRaw]
```

```
{}
```

Colorado (CO) is the only contiguous non-visited state.
In sentence #65, we are told that "each **card** is a **door**". This means that we may combine the two words, as they are the same. The state is **CO**.
Then the sentence ends with "woman", which allows us to select **NM**, contiguous with **CO**.
So we select two states: **CO-NM**

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE-MO-TN-AR-OK-KS-CO-NM

67 – 68

These two sentences are best taken together.
Possible non-visited states from New Mexico (NM) are AZ, UT, TX.
In sentence #67 Amanda "did not know which card to **pick**". A **pick** is used to play a musical instrument. And **UT** is actually a musical note (equivalent to DO). We could conclude that the state we want is Utah (UT), but Amanda doesn't know and more importantly, in sentence #68, she **passed** to the next one!
In conclusion, she passes through Utah (UT). This will be equivalent to traversing Utah without visiting its capital. **UT will not be added to our Current Journey**. We will however add UT to our list of abbreviations in our function call so this state doesn't come up again.
We run the function with sentence 67 from UTAH (UT) to determine where she ends up.

```
processStateSentences[67,
 getPossibleStates["UT", {"NJ", "DE", "MD", "WV", "OH", "MI", "MN", "ND", "SD",
   "IA", "NE", "MO", "TN", "AR", "OK", "KS", "CO", "NM", "UT"}], fortuneRaw]
```

```
{ID}
```

Current Journey: NJ-DE-MD-WV-OH-MI-MN-ND-SD-IA-NE-MO-TN-AR-OK-KS-CO-NM-ID

69

```
processStateSentences[69,
 getPossibleStates["ID", {"NJ", "DE", "MD", "WV", "OH", "MI", "MN", "ND", "SD",
   "IA", "NE", "MO", "TN", "AR", "OK", "KS", "CO", "NM", "UT", "ID"}], fortuneRaw]
```

```
{OR}
```

In sentence#69, Oregon (**OR**) is found in “room”, but also California (**CA**) in “change”, a state contiguous to Oregon.
We add **OR-CA**

Final Fortune Journey States : NJ – DE – MD – WV – OH – MI –
MN – ND – SD – IA – NE – MO – TN – AR – OK – KS – CO – NM – ID – OR – CA

We have now visited 21 states, as predicted by the the 21 spots on the dice. We are done. Indeed, the next two sentences are:

71→The woman said not a word more
73→Whatever had happened, it was over now.

The sentence following these two is also interesting:

75→”Clutching her card to her breast, she hurried from the room.”

A **clutch** is a device in some vehicles that allows a driver to shift gears. It seems that Amanda will need a car to finish her Journey.

ENDING VERSION 2 MODIFICATIONS

We may now convert our Journey’s 21 states to their capitals and get a visual representation of our trip across the United States.
Note that we are using straight lines only, to show the order in which the state capitals were visited (red dots), not the path followed between two successive points. For example, the inverted U over Wisconsin (between Michigan and Minnesota) is not depicted. Instead, we show a straight line going through Wisconsin.

```
fortuneCapitals = {"Trenton", "Dover", "Annapolis", "Charleston", "Columbus",
   "Lansing", "SaintPaul", "Bismark", "Pierre", "Des Moines", "Lincoln",
   "Jefferson City", "Nashville", "LittleRock", "Oklahoma City",
   "Topeka", "Denver", "Santa Fe", "Boise", "Salem", "Sacramento"};
```

```
fortuneCapitalEntities = {Trenton CITY ✓, Dover CITY ✓,
    Annapolis CITY ✓, Charleston CITY ··· ✓,
    Columbus CITY ✓, Lansing CITY ✓, Saint Paul CITY ✓,
    Bismarck CITY ✓, Pierre CITY ✓, Des Moines CITY ··· ✓,
    Lincoln CITY ✓, Jefferson City CITY ✓, Nashville CITY ✓, Little Rock CITY ✓,
    Oklahoma City CITY ··· ✓, Topeka CITY ✓,
    Denver CITY ··· ✓, Santa Fe CITY ✓, Boise CITY ✓,
    Salem CITY ✓, Sacramento CITY ✓
  };
```

```
GeoGraphics[{EdgeForm[Black], FaceForm[Red], GeoPath[{fortuneCapitalEntities}],
  Red, PointSize[Medium], (Point[GeoPosition@#] & /@ fortuneCapitalEntities)}]
```

Chapter 7 - ROAD

Our Journey started in Chapter 5, but has not yet been completed. The end of the Journey is found in

ROAD.

STARTING VERSION 2 MODIFICATIONS

The text of ROAD is separated into two sections by a long and narrow horizontal image going across the four pages of the chapter. A sentence describes the presence of two sides: “On one side of the door .... On the other side was the road”, “It was a matter of choosing the right direction”. We also have the images of the car going in one direction, then in the other direction.

This is because we will be adding more cities to our Journey on both the Eastern and Western portion of the trip

In Version 1, lacking a “sentence selector”, we used different areas of the chapter (above and below the horizontal image) to look for applicable clues. In the present version, a selector was found, so we can use a strict procedure to generate states without relying on arbitrary selection of paragraphs. We will still use the two-letter code for states found in a single word to identify a state (as we did before in FORTUNE), but selecting the sentences to be considered will change dramatically.

We start with the text of the chapter and numbering of sentences.

```
roadRaw =
  "Enchanted with the city, Amanda combed it endlessly for clues. But as week
    after week went by, the enchantment waned. Soon she began to feel
    awash in the babel of crowds and cars and to feel trapped in air so
    hazy she could barely see through it. Now she had a book and carried
    it with her everywhere. It was bound in maroon leather and had thick
    paper pages, on which she wrote every important thing that she saw. Any
    sign of a horse. Any coincidence or paradox. Any puzzling thing. All
    were recorded in the book. Every night, as Amanda sat in her room at
    the inn, she looked back over what she had written. She noticed that
    she had begun to write more and more about The Journey, and where it
    might lead her. She was beginning to understand that it was up to her
    to find a path. If she chose this fork, or that turn, it could take
    her in any direction. It was a matter of choosing the right direction,
    and more and more she felt that the right direction was away from the
    city. But without her stallion to carry her away, The Journey seemed
    impossible. Then she found the car. It was older than she, sparkling
    white and in perfect condition. On its grille, leading the way, was
    the figure of a running horse. She hesitated, unsure of what to do.
    She looked inside and saw the note on the driver's seat. It read,
    Start me to your Journey's End. She sat behind the wheel and turned
    the key, and it was like opening a door. On one side of the door was
```

```
    the city and all her failures. On the other side was the road. The
    road was a landscape unlike any other, a blend of wind and clouds,
    and the sun flashing through passing trees. It was a landscape of
    constant change, as if the road were flowing through her, and she
    flying above it. The road led her along the glimmering ocean, past
    plunging seabirds, and into the night. It took her inland, through
    many small towns. She stopped in each town and described her lost
    horse. No one thought it strange, and they took the time to listen,
    to think, and to tell her what they had heard. Many times she heard
    the rumor of a phantom horse. It roamed somewhere to the east, they
    told her in one small village. A magnificient Arabian stallion, they
    told her in another. In a third town some said the stallion had lost
    his master. Then she heard the stallion lived alone in the desert,
    drawn to the heatand sand by a thousand years of blood memory. She
    stopped to buy apples from an old man at the side of the road, and
    he told her about an old Indian corral on the edge of the desert. A
    horse was sometimes seen there, a stallion. He came to the corral,
    the story went, looking for his missing master, the master he still
    loved. She knew the stallion was her Treasure. The air turned hot
    as she drove eastward. Unlike the city, her it was dry and clear
    with sunrises that were violent in color. She had been told that the
    corral was invisible from the road. So she turned onto an unmarked
    trail and followed dusty, rutted tracks leading to the edge of the
    desert. And there it was. The car's engine sputtered to a stop.
    Amanda felt a strange presence in the air, like silent thunder.
    She was certain Treasure was there. The corral had been formed of
    weathered split-rails in the shape of a figure eight, with an extra
    circle in its center. She ran to the fence and climbed up to see into
    the corral. It was empty. She hurried to the gate, unlatched the
    rusty chain, and entered the corral itself. She turned and looked
    for life on any kind. But there was nothing. She knew she should
    wait for him to come to her, but she could not. Her eyes caught the
    line of a great black bird that circled and soared above her. In
    desperation, she followed his shadow into the shimmering desert.";
```

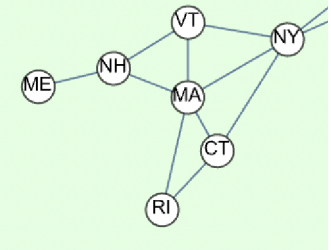
```
Column@Normal@(AssociationThread[
    Range[Length[TextSentences[roadRaw]]], TextSentences[roadRaw]])
```

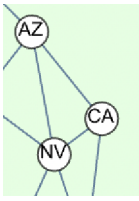
```
1 → Enchanted with the city, Amanda combed it endlessly for clues.
2 → But as week after week went by, the enchantment waned.
3 → Soon she began to feel awash in the babel of crowds and cars and
    to feel trapped in air so hazy she could barely see through it.
4 → Now she had a book and carried it with her everywhere.
```

```
5 → It was bound in maroon leather and had thick paper
   pages, on which she wrote every important thing that she saw.
6 → Any sign of a horse.
7 → Any coincidence or paradox.
8 → Any puzzling thing.
9 → All were recorded in the book.
10 → Every night, as Amanda sat in her room
   at the inn, she looked back over what she had written.
11 → She noticed that she had begun to write
   more and more about The Journey, and where it might lead her.
12 → She was beginning to understand that it was up to her to find a path.
13 → If she chose this fork, or that turn, it could take her in any direction.
14 → It was a matter of choosing the right direction, and more and
   more she felt that the right direction was away from the city.
15 → But without her stallion to carry her away, The Journey seemed impossible.
16 → Then she found the car.
17 → It was older than she, sparkling white and in perfect condition.
18 → On its grille, leading the way, was the figure of a running horse.
19 → She hesitated, unsure of what to do.
20 → She looked inside and saw the note on the driver's seat.
21 → It read, Start me to your Journey's End.
22 → She sat behind the wheel and turned the key, and it was like opening a door.
23 → On one side of the door was the city and all her failures.
24 → On the other side was the road.
25 → The road was a landscape unlike any other, a blend of
   wind and clouds, and the sun flashing through passing trees.
26 → It was a landscape of constant change, as if
   the road were flowing through her, and she flying above it.
27 → The road led her along the glimmering
   ocean, past plunging seabirds, and into the night.
28 → It took her inland, through many small towns.
29 → She stopped in each town and described her lost horse.
30 → No one thought it strange, and they took the
   time to listen, to think, and to tell her what they had heard.
31 → Many times she heard the rumor of a phantom horse.
32 → It roamed somewhere to the east, they told her in one small village.
33 → A magnificient Arabian stallion, they told her in another.
34 → In a third town some said the stallion had lost his master.
35 → Then she heard the stallion lived alone in the desert,
   drawn to the heatand sand by a thousand years of blood memory.
36 →
 She stopped to buy apples from an old man at the side of the road, and he told
   her about an old Indian corral on the edge of the desert.
37 → A horse was sometimes seen there, a stallion.
38 → He came to the corral, the story went,
   looking for his missing master, the master he still loved.
39 → She knew the stallion was her Treasure.
```

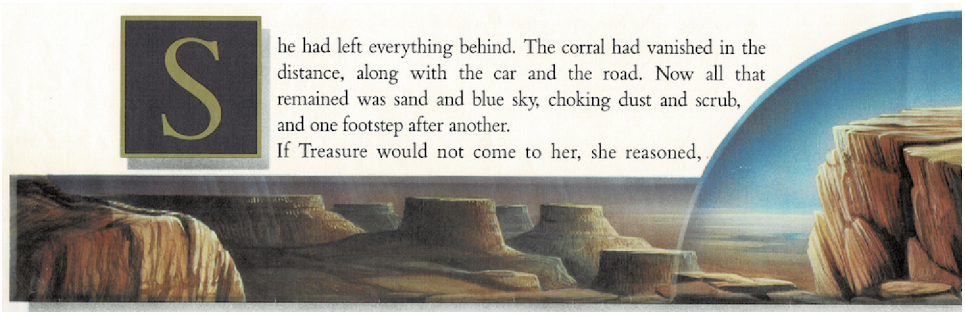
```
40 → The air turned hot as she drove eastward.
41 → Unlike the city, her it was
     dry and clear with sunrises that were violent in color.
42 → She had been told that the corral was invisible from the road.
43 → So she turned onto an unmarked trail and followed
     dusty, rutted tracks leading to the edge of the desert.
44 → And there it was.
45 → The car's engine sputtered to a stop.
46 → Amanda felt a strange presence in the air, like silent thunder.
47 → She was certain Treasure was there.
48 → The corral had been formed of weathered split-rails in
     the shape of a figure eight, with an extra circle in its center.
49 → She ran to the fence and climbed up to see into the corral.
50 → It was empty.
51 → She hurried to the gate,
     unlatched the rusty chain, and entered the corral itself.
52 → She turned and looked for life on any kind.
53 → But there was nothing.
54 → She knew she should wait for him to come to her, but she could not.
55 → Her eyes caught the line of
     a great black bird that circled and soared above her.
56 → In desperation, she followed his shadow into the shimmering desert.
```

56 sentences is obviously more than we should need. As seen on the contiguity graph, there are only a few states left which could be added to California on the western side (**NV, AZ**) and to New Jersey on the Eastern Side (**ME, NH, VT, MA, RI, CT, NY**). Note that NY does connect to our starting point NJ.

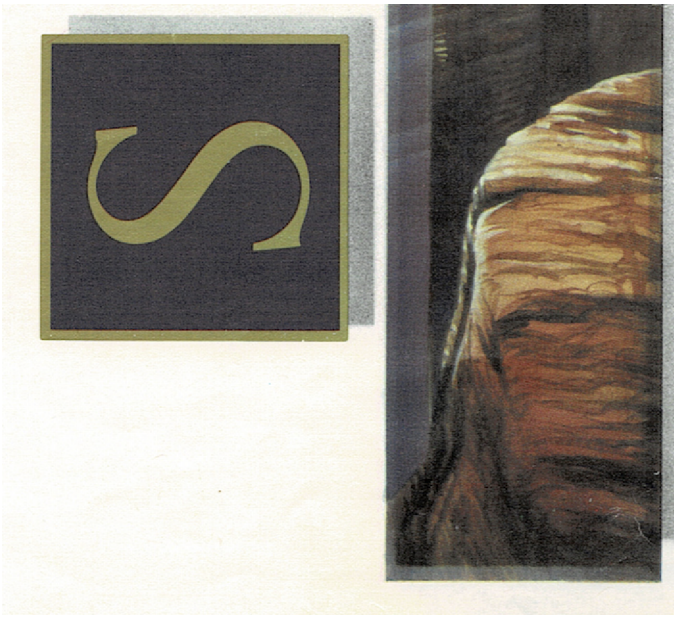




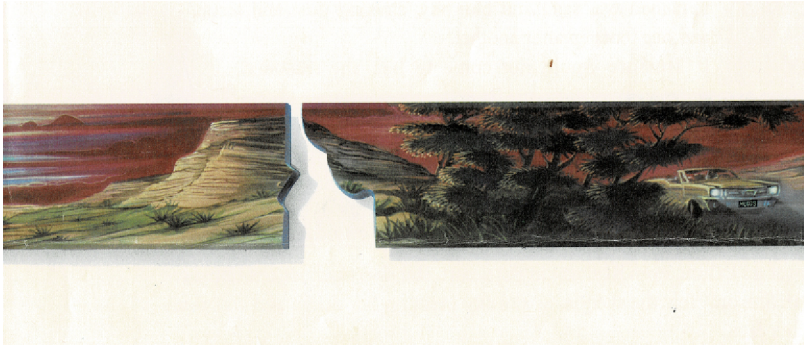
Sentence #26 is a clue that ROAD is not to be read one sentence at a time. We need to “fly above the road”. This implies that we must have some numbers indicating where to jump next.
In Sentence #22, the **key** is turned. What is this key? We seem to have an excess of keys, with THE MAP IS THE KEY... and the musical key of E-major. This time, we will introduce the **shape of a key**.
This key is actually quite obvious as an image in WILDERNESS.



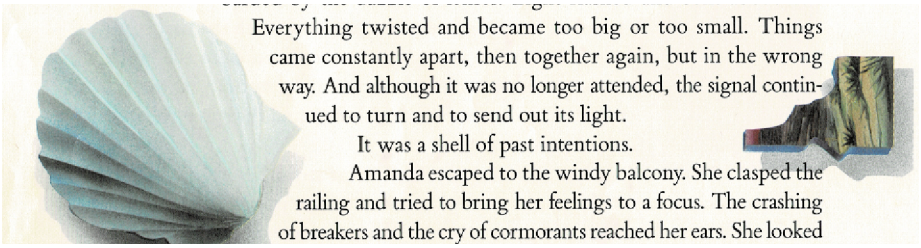
We have the bow of the key on the right, with the shoulder. Then the mountains with flat tops playing the role of the cuts of a modern high security key. But the big **S** could illustrate an old fashioned key. There is a face in the rock on the left. This is easier to see when the page is vertical. This is Amanda being turned to stone, where the rocks “towered **above** her”.



The **S** seen this way looks like **waves**. This is the **S** of the word **STATE** in the text IGNORE STATE SHAPES. We already know that states are the key to The Journey. Interestingly, there is a state shape in ROAD (page 36). Is this the key?



This is the state of IDAHO. The missing piece in this image is found is SIGNAL.


Everything twisted and became too big or too small. Things came constantly apart, then together again, but in the wrong way. And although it was no longer attended, the signal continued to turn and to send out its light.

It was a shell of past intentions.

Amanda escaped to the windy balcony. She clasped the railing and tried to bring her feelings to a focus. The crashing of breakers and the cry of cormorants reached her ears. She looked

This missing piece, which does look more or less like a key, points to a specific sentence: **IT WAS A SHELL OF PAST INTENTIONS**.

What is immediately interesting about THE SIGNAL is the last sentence: “For now she could see a great distance and in many directions”. This is what we are after in ROAD! A way to fly above the road.

THE SIGNAL has an image of a light house with squares on the border, begging to be filled with letters. There are 22 such squares.

IT WAS A SHELL OF PAST INTENTIONS has 27 letters. We would need to remove 5 letters. This is precisely what happens if we HIDE A O (IDAHO!!). Removing the letters A and O leaves us with 22 letters. Starting in the bottom right corner (because it is different from the other corners and appears to show 0), we write our 22-letter text.

The four corners highlighted in blue now spell **SINE**, which is a perfect conclusion for THE SIGNAL, with all the references to waves: the ocean, light, lenses, shapes like spirals, circles, etc. We may be tempted here to take a mathematical direction with sine and cosine applied somehow to degrees, but that would probably make the puzzle too complex.

If **sine** is the key, then Sentence #22 has a certain mathematical logic in it: " She sat behind the wheel and turned the key". SINE is associated with circles (and wheel).

We will take the letters **S-I-N-E** and see how these could be applied to our sentences in ROAD, as a **selector** of sentences to process.

We start by calculating the total number of letters **S-I-N-E** in each of the 56 sentences.

```
tk = Total@KeyTake[LetterCounts[#, IgnoreCase → True], {"s", "i", "n", "e"}] & /@
    TextSentences[roadRaw];
mcc = Multicolumn[
  Normal@(AssociationThread[Range[Length[TextSentences[roadRaw]]], tk]),
  2, Spacings → 2]
```

```
1 → 17     29 → 17
2 → 16     30 → 23
3 → 31     31 → 11
4 → 13     32 → 18
5 → 27     33 → 17
6 → 6      34 → 17
7 → 7      35 → 30
8 → 5      36 → 29
9 → 7      37 → 17
10 → 23    38 → 25
11 → 24    39 → 13
12 → 18    40 → 10
13 → 18    41 → 24
14 → 28    42 → 14
15 → 18    43 → 25
16 → 6     44 → 5
17 → 20    45 → 11
18 → 19    46 → 22
19 → 9     47 → 12
20 → 20    48 → 33
21 → 9     49 → 16
22 → 23    50 → 3
23 → 15    51 → 21
24 → 8     52 → 12
25 → 30    53 → 6
26 → 25    54 → 14
27 → 25    55 → 17
28 → 9     56 → 24
```

We will use these numbers to determine how far to jump, or fly above the road, after processing a sentence. This is a different approach than the one used in FORTUNE.

Sentence #21 provides us with a starting point: “It read, **Start me** to your Jour**n**e**y**’s End.” It tells us to start with state **ME** and end when **NY** is reached.

We now use the same function **processStateSentences** to process the sentences looking for states. However, since we are potentially adding states on both the western and eastern section we will directly provide the contiguous possible states on both the Eastern and Western sides.

As before, sentence numbers are shown in a box. When we leave a sentence we jump by the **S-I-N-E** number of this sentence to find the next one to consider.

21

```
processStateSentences[21, {"ME", "AZ", "NV"}, roadRaw]
```

```
{ME}
```

Our starting point is given (ME) and is the only possible state on the eastern site.

Current Eastern Journey: ME

30 (21 + 9)

```
processStateSentences[30, {"NH", "AZ", "NV"}, roadRaw]
```

```
{NH}
```

Current Eastern Journey: ME-NH

53 (30 + 23)

```
processStateSentences[53, {"VT", "MA", "AZ", "NV"}, roadRaw]
```

```
{}
```

Note that sentence #53 is: “But there was nothing”. This is a confirmation of the procedure.

Current Eastern Journey: ME-NH

3 (53+6 = 3)

```
processStateSentences[3, {"VT", "MA", "AZ", "NV"}, roadRaw]
```

```
{AZ}
```

Current Western Journey: AZ

Current Eastern Journey: ME-NH

34 (3 + 31)

```
processStateSentences[34, {"VT", "MA", "NV"}, roadRaw]
```

```
{MA}
```

Current Western Journey: AZ

Current Eastern Journey: ME-NH-MA

51 (34 + 17)

```
processStateSentences[51, {"VT", "NI", "CT", "RI", "NV"}, roadRaw]
```

```
{CT, RI}
```

This is an interesting sentence. In order, we get **RI** (from *hurried*), then **CT** (from *unlatched*). But there is more! It is the *rusty chain* that is unlatched, so we have to reattach it forming one word: *rustychain*. And this gives us **NY**, marking the end of the Eastern Journey and the end of The Journey since we were told to stop at NY.

Current Western Journey: AZ

Current Eastern Journey: ME-NH-MA-RI-CT-NY

As can be seen, having the proper selector for sentences makes looking for states very efficient.

Final Journey

We have added 7 states, for a total of 28 states in our Journey.

```
statesVisitedFinal = {"ME", "NH", "MA", "RI", "CT", "NY",
   "NJ", "DE", "MD", "WV", "OH", "MI", "MN", "ND", "SD", "IA", "NE",
   "MO", "TN", "AR", "OK", "KS", "CO", "NM", "ID", "OR", "CA", "AZ"};
Length[statesVisitedFinal]
```

```
28
```

We now change all these state abbreviations for state capitals.

```
capitalsVisited = {"Augusta", "Concord", "Boston", "Providence", "Hartford",
   "Albany", "Trenton", "Dover", "Annapolis", "Charleston", "Columbus",
   "Lansing", "SaintPaul", "Bismark", "Pierre", "Des Moines", "Lincoln",
   "Jefferson City", "Nashville", "LittleRock", "Oklahoma City", "Topeka",
   "Denver", "Santa Fe", "Boise", "Salem", "Sacramento", "Phoenix"};
```

Before going further, we need to consider the chapter **FISH**, which stands in the middle of The Journey, between FORTUNE and ROAD. The purpose of FISH is clearly to **cut** letters (the chef swinging “a huge wet knife”. This will be done in two places. One involves the Oriental chef, the other his wife.

The first cut has to do with the capital of New York state, which is is **ALBANY**.

In FISH we have:
- an Oriental chef with a knife, indicating that we need to cut on the eastern end of the Journey.
- the chef is dressed in white. The word **ALBA**, parts of **ALBANY**, the capital of **NY**, means **WHITE**.

- the chef wears “the thickest glasses she had ever seen”. This reference to a thick lens brings us to the image of the tarot cards in FORTUNE, viewed through such a thick lens, warping the image.
- Amanda is said to “remember what the dealer of cards had told her”, again a reference to the tarot cards.

In FORTUNE, we must focus on the cards touched by the fingers.
THE DEVIL is bad and represents what has to be removed or hidden. “Next to him, a woman grieved and hid her eyes”.
The WHEEL OF FORTUNE is referred to as a “turning waterwheel”. Wheel of Fortune was also a popular game show involving hidden letters. Four letters are hidden on this card: ROUE (French for Wheel). So we are looking for four letters representing the devil.
The answer to this puzzle is **BAAL**. But these letters are on a turning wheel. Rotating this wheel by 2 letters we get:

```
StringRotateLeft["BAAL", 2]
```

```
ALBA
```

ALBA is what has to be removed. It appears only in the capital of New York State, **ALBANY**. Removing these four letters, we are left with **NY**, making **New York City** the city to use for New York State instead of its capital.

The third card touched by a finger is LE SOLEIL (French for THE SUN). In the word SOLEIL, two letters are covered. Does this refer to the fact that we are left with the two letters NY? I am not sure. Another possibility is that “The sun rose in the seventh card and drenched the world in light” refers to THE SUN, a very successful **NEW YORK CITY** newspaper from 1833 to 1950, credited for “making news available

for the first time to working men and woman...", and therefore for drenching the world in light.

ENDING VERSION 2 MODIFICATIONS

We are now able to list all the cities visited, most of them state capitals:

```
citiesVisitedFinal = {"Augusta", "Concord", "Boston", "Providence", "Hartford",
   "New York City", "Trenton", "Dover", "Annapolis", "Charleston", "Columbus",
   "Lansing", "SaintPaul", "Bismark", "Pierre", "Des Moines", "Lincoln",
   "Jefferson City", "Nashville", "LittleRock", "Oklahoma City", "Topeka",
   "Denver", "Santa Fe", "Boise", "Salem", "Sacramento", "Phoenix"};
```

(Note that our final destination ends with X).

```
finalCitiesEntities = {
    Augusta CITY ✓, Concord CITY ••• ✓, Boston CITY ••• ✓,
    Providence CITY ••• ✓, Hartford CITY ••• ✓, New York City CITY ••• ✓,
    Trenton CITY ✓, Dover CITY ✓,
    Annapolis CITY ✓, Charleston CITY ••• ✓,
    Columbus CITY ✓, Lansing CITY ✓, Saint Paul CITY ✓,
    Bismarck CITY ✓, Pierre CITY ✓, Des Moines CITY ••• ✓,
    Lincoln CITY ✓, Jefferson City CITY ✓, Nashville CITY ✓, Little Rock CITY ✓,
    Oklahoma City CITY ••• ✓, Topeka CITY ✓,
    Denver CITY ••• ✓, Santa Fe CITY ✓, Boise CITY ✓,
    Salem CITY ✓, Sacramento CITY ✓, Phoenix CITY ••• ✓
  };
```

```
GeoGraphics[{EdgeForm[Black], FaceForm[Red], GeoPath[{finalCitiesEntities}],
  Red, PointSize[Medium], (Point[GeoPosition@#] & /@ finalCitiesEntities)}]
```

The purpose of The Journey was to collect a string of letters. This string is made up of all the names of cities we visited.

```
citiesString = StringDelete[
  StringJoin@citiesVisitedFinal, Except[Alphabet[]], IgnoreCase → True]
```

```
AugustaConcordBostonProvidenceHartfordNewYorkCityTrentonDoverAnnapolisCharlesto
   nColumbusLansingSaintPaulBismarkPierreDesMoinesLincolnJeffersonCityNashville
   LittleRockOklahomaCityTopekaDenverSantaFeBoiseSalemSacramentoPhoenix
```

```
StringLength@citiesString
```

```
223
```

We have 223 characters, much longer than the 100 characters required to fill the map border.
The man who is also a bird showed up while we were traveling around the country, particularly in FORTUNE, where we were warned about him by the woman. We also have music (the flute) playing in FISH. We will cut the letters **C D F G** from our string.

```
citiesStringCDFG =
 StringDelete[citiesString, {"c", "d", "f", "g"}, IgnoreCase → True]
```

```
AuustaonorBostonProvieneHartorNewYorkityTrentonoverAnnapolisharlestonolumbusLan
   sinSaintPaulBismarkPierreesMoinesLinolnJeersonityNashvilleLittleRokOklahomai
   tyTopekaenverSantaeBoiseSalemSaramentoPhoenix
```

```
StringLength@citiesStringCDFG
```

```
200
```

We now have 200 letters, twice what is needed. This is a typical situation where either odd or even characters will be used. We refer again to chapter 6 (FISH), sitting between FORTUNE and ROAD, in the middle of our Journey. It is obviously about cutting letters. The chef prepares something and Amanda forces herself to stay because “Clues could be anywhere. **Even** here.” But she says NO to what the chef offers. She then turns to the the chef’s wife (his other **half**). The woman ends up writing the letters on **parchment** (the map). In any case, taking the **odd** position letters and discarding the **even** position letters provides a powerful confirmation: we find **AUTO**, the car leading the way! This is the road!

```
journeyStringLetters =
 #[[1]] & /@ Partition[ToUpperCase@Characters@StringJoin[citiesStringCDFG], 2]
```

```
{A, U, T, O, O, B, S, O, P, O, I, N, H, R, O, N, W, O, K, T, T, E, T, N,
 V, R, N, A, O, I, H, R, E, T, N, L, M, U, L, N, I, S, I, T, A, L, I, M, R,
 P, E, R, E, M, I, E, L, N, L, J, E, S, N, T, N, S, V, L, E, I, T, E, O, O,
 L, H, M, I, Y, O, E, A, N, E, S, N, A, B, I, E, A, E, S, R, M, N, O, H, E, I}
```

```
Length[journeyStringLetters]
```

```
100
```

We now need to copy these 100 letters on the map, clockwise (direction of travel of the four horses). But where do we start? There are 100 possible starting slots on the map border. Unable to determine the starting location, I had to try them all, hoping one will somehow speak to me. And one did. The following is a mockup of the actual map.

```
rowJourneyRaw =
  {"LMULNISITALIMRPEREMIELNLJE", "N_S", "T_N", "E_T", "R_N", "H_S", "I_V", "O_L",
   "A_E", "N_I", "R_T", "V_E", "N_O", "T_O", "E_L", "T_H", "T_M", "K_I", "O_Y",
   "W_O", "N_E", "O_A", "R_N", "H_E", "N_S", "IOPOSBOOTUAIEHONMRSEAEIBAN"};
rowJourney =
  Characters@StringReplace[#, "_" → StringJoin@Table[" ", 24]] & /@ rowJourneyRaw;
Grid[rowJourney,
 Frame → All, ItemSize → {1, 1}, Background →
  {None, None, {{6, 1} → Cyan, {6, 26} → Cyan, {1, 13} → Cyan, {26, 13} → Cyan,
    {1, 1} → LightOrange, {1, 26} → LightOrange, {26, 1} → LightOrange,
    {26, 26} → LightOrange, {{2, 25}, {2, 25}} → LightGray, {5 + 1, 12 + 1} → White}}]
```

L	M	U	L	N	I	S	I	T	A	L	I	M	R	P	E	R	E	M	I	E	L	N	L	J	E
N																									S
T																									N
E																									T
R																									N
H																									S
I																									V
O																									L
A																									E
N																									I
R																									T
V																									E
N																									O
T																									O
E																									L
T																									H
T																									M
K																									I
O																									Y
W																									O
N																									E
O																									A
R																									N
H																									E
N																									S
I	O	P	O	S	B	O	O	T	U	A	I	E	H	O	N	M	R	S	E	A	E	I	B	A	N

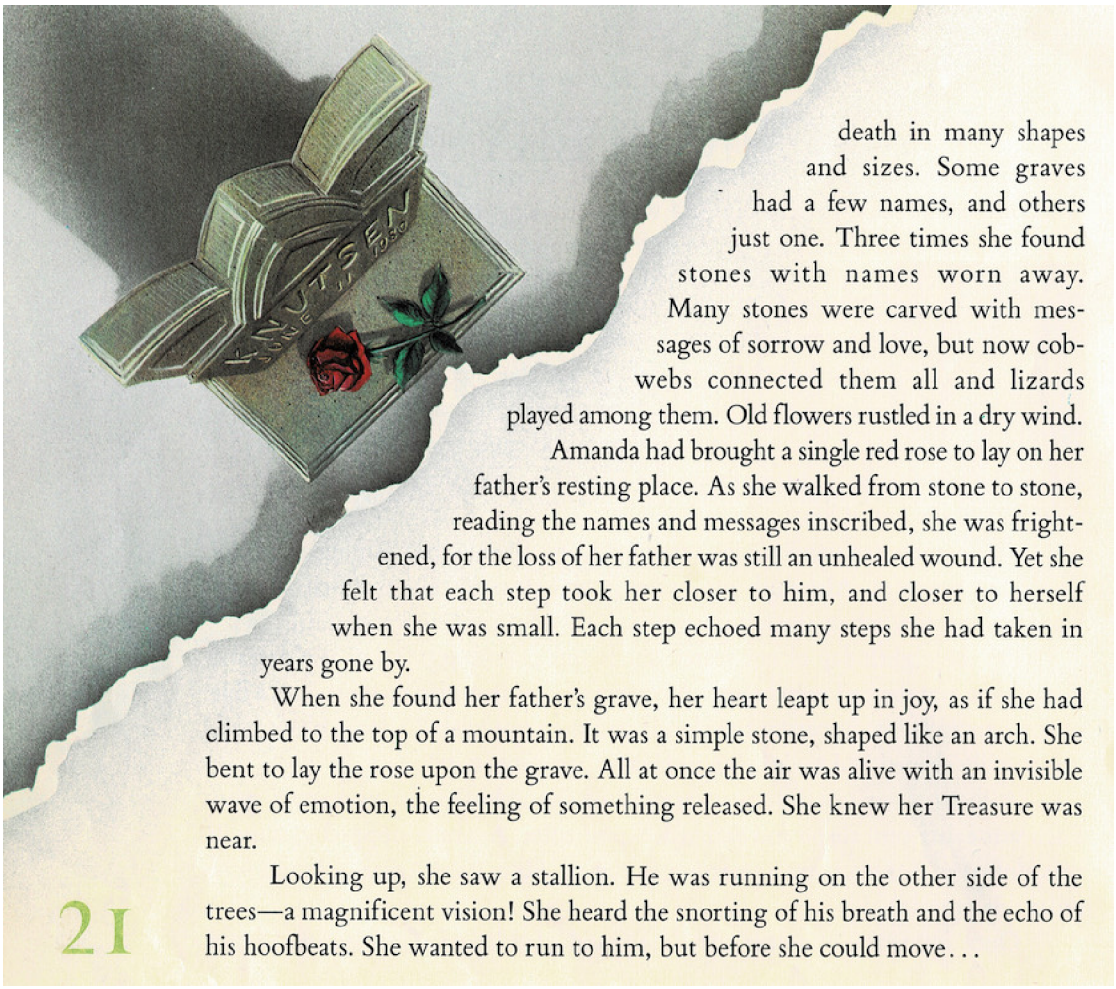
The white square is the position of the X on the map. The cyan color indicates the position of horses. These four letters spell the word MESH, suggesting a 24x24 mesh or grid in the middle of the map. This also suggests that this grid must be filled with more letters.

The four corners spell the word NILE, which seems to reference chapter 15 (RIVER). The image in RIVER does show some Egyptian hieroglyphics. The corners on the real map are triangular and point toward

the center circle, as if marking the four directions. We can already guess what it means: RIVER will move some token in different directions on the grid of fixed characters, while horses will move on the Journey's characters on the border...but we are getting ahead of ourselves. We must find the letters to fill the grid. Where do we go to read fixed characters on stones? The CEMETERY.

CEMETERY

What immediately attracts attention in CEMETERY is page 21. As we have seen in COURTYARD, this type of triangular view, seemingly showing a page underneath the text page, indicates that the text on this particular page is the important text. Note that on page 20 we have the sentence “First steps can be very painful”. Indeed, our first word is death.



We need 24 x 24 = 576 letters to fill the grid in the center of the map. We will start by computing what we have here.

```
(* The text on page 21 *)

cemeteryP21Raw =
  "death in many shapes and sizes. Some graves had a few names, and others
    just one. Three times she found stones with names worn away. Many
    stones were carved with messages of sorrow and love, but now cobwebs
    connected them all and lizards played among them. Old flowers rustled
    in a dry wind. Amanda had brought a single red rose to lay on her
    father's resting place. As she walked from stone to stone, reading
    the names and messages inscribed, she was frightened, for the loss of
    her father was still an unhealed wound. Yet she felt that each step
    took her closer to him, and closer to herself when she was small.
    Each step echoed many steps she had taken in the years gone by. When
    she found her father's grave, her heart leapt up in joy, as if she
    had climbed to the top of a mountain. It was a simple stone, shaped
    like an arch. She bent to lay the rose upon the grave. All at once
    the air was alive with an invisible wave of emotion, the feeling
    of something released. She knew her Treasure was near. Looking up,
    she saw a stallion. He was running on the other side of the trees a
    magnificient vision. She heard the snorting of his breath and the echo
    of his hoofbeats. She wanted to run to him, but before she could move.";
```

```
(* The text of page 21 split into sentences *)

cemeterySentencesP21Raw = TextSentences[cemeteryP21Raw]
```

```
{death in many shapes and sizes.,
 Some graves had a few names, and others just one.,
 Three times she found stones with names worn away.,
 Many stones were carved with messages of sorrow and love, but
   now cobwebs connected them all and lizards played among them.,
 Old flowers rustled in a dry wind.,
 Amanda had brought a single red rose to lay on her father's resting place.,
 As she walked from stone to stone, reading the
   names and messages inscribed, she was frightened, for
   the loss of her father was still an unhealed wound.,
 Yet she felt that each step took her closer to him, and
   closer to herself when she was small.,
 Each step echoed many steps she had taken in the years gone by.,
 When she found her father's grave, her heart leapt up
   in joy, as if she had climbed to the top of a mountain.,
 It was a simple stone, shaped like an arch.,
 She bent to lay the rose upon the grave.,
 All at once the air was alive with an invisible
   wave of emotion, the feeling of something released.,
 She knew her Treasure was near., Looking up, she saw a stallion.,
 He was running on the other side of the trees a magnificient vision.,
 She heard the snorting of his breath and the echo of his hoofbeats.,
 She wanted to run to him, but before she could move.}
```

```
Column@Normal@(AssociationThread[Range[Length[TextSentences[cemeteryP21Raw]]],
    TextSentences[cemeteryP21Raw]])
```

```
1 → death in many shapes and sizes.
2 → Some graves had a few names, and others just one.
3 → Three times she found stones with names worn away.
4 → Many stones were carved with messages of sorrow and love, but
   now cobwebs connected them all and lizards played among them.
5 → Old flowers rustled in a dry wind.
6 → Amanda had brought a single red rose to lay on her father's resting place.
7 → As she walked from stone to stone, reading the names and messages inscribed,
   she was frightened, for the loss of her father was still an unhealed wound.
8 → Yet she felt that each step took her
   closer to him, and closer to herself when she was small.
9 → Each step echoed many steps she had taken in the years gone by.
10 → When she found her father's grave, her heart leapt
   up in joy, as if she had climbed to the top of a mountain.
11 → It was a simple stone, shaped like an arch.
12 → She bent to lay the rose upon the grave.
13 → All at once the air was alive with an
   invisible wave of emotion, the feeling of something released.
14 → She knew her Treasure was near.
15 → Looking up, she saw a stallion.
16 → He was running on the other side of the trees a magnificient vision.
17 → She heard the snorting of his breath and the echo of his hoofbeats.
18 → She wanted to run to him, but before she could move.
```

We have numbered sentences, but here we must also deal with words and letters.

```
(* The text of page 21 split into words *)

cemeteryWordsP21Raw = TextWords[cemeteryP21Raw]
```

```
{death, in, many, shapes, and, sizes, Some, graves, had, a, few, names, and,
 others, just, one, Three, times, she, found, stones, with, names, worn,
 away, Many, stones, were, carved, with, messages, of, sorrow, and, love,
 but, now, cobwebs, connected, them, all, and, lizards, played, among, them,
 Old, flowers, rustled, in, a, dry, wind, Amanda, had, brought, a, single,
 red, rose, to, lay, on, her, father's, resting, place, As, she, walked, from,
 stone, to, stone, reading, the, names, and, messages, inscribed, she, was,
 frightened, for, the, loss, of, her, father, was, still, an, unhealed, wound,
 Yet, she, felt, that, each, step, took, her, closer, to, him, and, closer, to,
 herself, when, she, was, small, Each, step, echoed, many, steps, she, had,
 taken, in, the, years, gone, by, When, she, found, her, father's, grave, her,
 heart, leapt, up, in, joy, as, if, she, had, climbed, to, the, top, of, a,
 mountain, It, was, a, simple, stone, shaped, like, an, arch, She, bent, to,
 lay, the, rose, upon, the, grave, All, at, once, the, air, was, alive, with,
 an, invisible, wave, of, emotion, the, feeling, of, something, released,
 She, knew, her, Treasure, was, near, Looking, up, she, saw, a, stallion,
 He, was, running, on, the, other, side, of, the, trees, a, magnificient,
 vision, She, heard, the, snorting, of, his, breath, and, the, echo, of, his,
 hoofbeats, She, wanted, to, run, to, him, but, before, she, could, move}
```

```
StringLength[StringJoin[cemeteryWordsP21Raw]]
```

```
963
```

We have almost twice as many letters as we need to fill the map. We need to examine the sentences for clues and make modifications accordingly.

Sentence numbers are framed below. Interpretation refers to the text of these sentences.

1

“Death” of some words is required. This will depend on the size of the words.

3

This sentence indicates that words containing a number of letters that is a multiple of three will be removed. But there are some specific rules. The word “single” in line 6 is considered one unit, not to be analyzed as a multiple of 3. It will not be removed even if it contains 6 letters.

```
cemeteryWordsP21Mod =
 Select[cemeteryWordsP21Raw, If[# == "single", True, Mod[StringLength[#], 3] ≠ 0] &]
```

```
{death, in, many, sizes, Some, a, names, just, Three, times, found, with, names,
 worn, away, Many, were, with, messages, of, love, cobwebs, them, lizards,
 among, them, flowers, rustled, in, a, wind, brought, a, single, rose, to,
 on, father's, resting, place, As, from, stone, to, stone, reading, names,
 messages, frightened, loss, of, still, an, unhealed, wound, felt, that, each,
 step, took, to, to, herself, when, small, Each, step, many, steps, taken,
 in, years, gone, by, When, found, father's, grave, heart, leapt, up, in, as,
 if, climbed, to, of, a, mountain, It, a, stone, like, an, arch, bent, to,
 rose, upon, grave, at, once, alive, with, an, wave, of, emotion, feeling, of,
 released, knew, Treasure, near, Looking, up, a, stallion, He, running, on,
 other, side, of, trees, a, heard, snorting, of, echo, of, to, to, could, move}
```

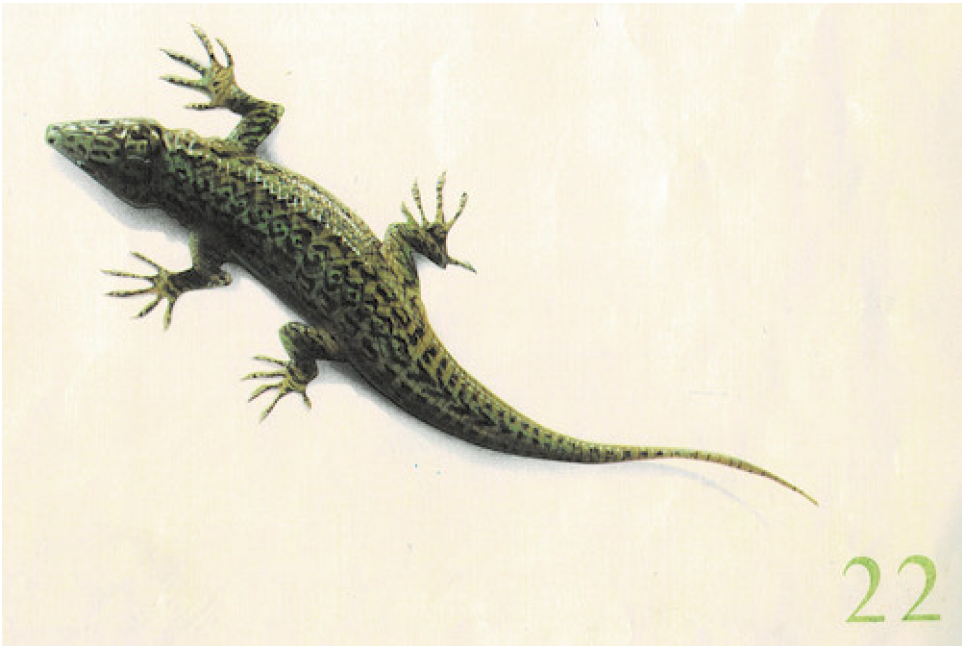
Another specific rule concerns the two words "**worn away**". Because wear caused the separation of the two words, we put them together as one word.

```
cemeteryWordsP21Mod2 =
 SequenceReplace[cemeteryWordsP21Mod, {{"worn", "away"} → "wornaway"}]
```

```
{death, in, many, sizes, Some, a, names, just, Three, times, found, with, names,
 wornaway, Many, were, with, messages, of, love, cobwebs, them, lizards,
 among, them, flowers, rustled, in, a, wind, brought, a, single, rose, to,
 on, father's, resting, place, As, from, stone, to, stone, reading, names,
 messages, frightened, loss, of, still, an, unhealed, wound, felt, that, each,
 step, took, to, to, herself, when, small, Each, step, many, steps, taken,
 in, years, gone, by, When, found, father's, grave, heart, leapt, up, in, as,
 if, climbed, to, of, a, mountain, It, a, stone, like, an, arch, bent, to,
 rose, upon, grave, at, once, alive, with, an, wave, of, emotion, feeling, of,
 released, knew, Treasure, near, Looking, up, a, stallion, He, running, on,
 other, side, of, trees, a, heard, snorting, of, echo, of, to, to, could, move}
```

4

We are told that "lizards play" among the stones. This indicates that the word **lizards** should move to a new position. But where? To find out we have to examine the lizard on page 22.



The lizard is currently located at position 23 in our list. Here we have its tail pointing to 22. Should we move it to position 22? No. This would be neglecting another important clue: the emphasis on the fingers of the animal. There are 20 fingers. The fingers are the **base** of the animal. So 22 is correct, but it is expressed in base 20. The number we are looking for requires converting to base 10.

```
FromDigits[{2, 2}, 20]
```

```
42
```

So we have to move “lizards” to position 42 in our list.

```
cemeteryWordsP21Mod3 = Insert[Delete[cemeteryWordsP21Mod2, 23], "lizards", 42]
```

```
{death, in, many, sizes, Some, a, names, just, Three, times, found, with, names,
 wornaway, Many, were, with, messages, of, love, cobwebs, them, among, them,
 flowers, rustled, in, a, wind, brought, a, single, rose, to, on, father's,
 resting, place, As, from, stone, lizards, to, stone, reading, names, messages,
 frightened, loss, of, still, an, unhealed, wound, felt, that, each, step,
 took, to, to, herself, when, small, Each, step, many, steps, taken, in,
 years, gone, by, When, found, father's, grave, heart, leapt, up, in, as, if,
 climbed, to, of, a, mountain, It, a, stone, like, an, arch, bent, to, rose,
 upon, grave, at, once, alive, with, an, wave, of, emotion, feeling, of,
 released, knew, Treasure, near, Looking, up, a, stallion, He, running, on,
 other, side, of, trees, a, heard, snorting, of, echo, of, to, to, could, move}
```

We need to delete the non-letter symbols (‘).

```
cemeteryWordsP21Mod4 = StringDelete[#, "'"] & /@ cemeteryWordsP21Mod3
```

```
{death, in, many, sizes, Some, a, names, just, Three, times, found, with, names,
 wornaway, Many, were, with, messages, of, love, cobwebs, them, among, them,
 flowers, rustled, in, a, wind, brought, a, single, rose, to, on, fathers,
 resting, place, As, from, stone, lizards, to, stone, reading, names, messages,
 frightened, loss, of, still, an, unhealed, wound, felt, that, each, step,
 took, to, to, herself, when, small, Each, step, many, steps, taken, in,
 years, gone, by, When, found, fathers, grave, heart, leapt, up, in, as, if,
 climbed, to, of, a, mountain, It, a, stone, like, an, arch, bent, to, rose,
 upon, grave, at, once, alive, with, an, wave, of, emotion, feeling, of,
 released, knew, Treasure, near, Looking, up, a, stallion, He, running, on,
 other, side, of, trees, a, heard, snorting, of, echo, of, to, to, could, move}
```

Let' s see how many letters we have now.

```
StringLength@StringJoin@cemeteryWordsP21Mod4
```

```
565
```

We are just missing 11 letters... exactly the number of letters on the father's grave: **KNUTSEN JUNE**. And these happen to be two **names**, exactly what is required as we are supposed to "read the **names** and messages...".

7	10	11	12	13

Sentence #7 tells us to write the two names read on the grave in the "**unhealed wound**".
Sentence #10 requires doing this by climbing to the top of a mountain. This "mountain" is the word WOUND that must be changed to MOUND.
This change from M to W is supported by sentences #11,#12 and #13. The ROSE is double-hue (red and green). The shape of the top of stone is a W. But in sentence #13 the "wave of emotion" corresponds to the middle part of this W. If this part is "waving" up and down it alternates with the shape of an M for the whole stone.
We implement these changes here and get a string of 576 letters to be written on the map.

```
cemeteryWordsP21Mod5 =
 ToUpperCase@StringJoin@(SequenceReplace[cemeteryWordsP21Mod4,
     {{"unhealed", "wound"} → "unhealedKNUTSENJUNEmound"}])
```

```
DEATHINMANYSIZESSOMEANAMESJUSTTHREETIMESFOUNDWITHNAMESWORNAWAYMANYWEREWITHMESSA
   GESOFLOVECOBWEBSTHEMAMONGTHEMFLOWERSRUSTLEDINAWINDBROUGHTASINGLEROSETOONFATH
   ERSRESTINGPLACEASFROMSTONELIZARDSTOSTONEREADINGNAMESMESSAGESFRIGHTENEDLOSSOF
   STILLANUNHEALEDKNUTSENJUNEMOUNDFELTTHATEACHSTEPTOOKTOTOHERSELFWHENSMALLEACHS
   TEPMANYSTEPSTAKENINYEARSGONEBYWHENFOUNDFATHERSGRAVEHEARTLEAPTUPINASIFCLIMBED
   TOOFAMOUNTAINITASTONELIKEANARCHBENTTOROSEUPONGRAVEATONCEALIVEWITHANWAVEOFEMO
   TIONFEELINGOFRELEASEDKNEWTREASURENEARLOOKINGUPASTALLIONHERUNNINGONOTHERSIDEO
   FTREESAHEARDSNORTINGOFECHOOFTOTOCOULDMOVE
```

```
StringLength[cemeteryWordsP21Mod5]
```

```
576
```

If we assume normal writing left to right and top to bottom, we can still start in four different positions on the grid, depending on what corner of the map is top-left. We have already shown the map in the correct orientation based on the fact that there is only one way to write our text with the letter M (shape of stone) falling on the X (the ROSE fixing directions) on the map.

```
topRowJourney = {"LMULNISITALIMRPEREMIELNLJE"};
sideRowsJourneyRaw = {"N_S", "T_N", "E_T", "R_N", "H_S",
    "I_V", "O_L", "A_E", "N_I", "R_T", "V_E", "N_O", "T_O", "E_L", "T_H",
    "T_M", "K_I", "O_Y", "W_O", "N_E", "O_A", "R_N", "H_E", "N_S"};
bottomRowJourney = {"IOPOSBOOTUAIEHONMRSEAEIBAN"};
cemeteryMap = Partition[Characters@cemeteryWordsP21Mod5, 24];
fullRows  = Join[topRowJourney,
    MapIndexed[StringReplace[#1, "_" → StringJoin@cemeteryMap[[#2]]] &,
     sideRowsJourneyRaw], bottomRowJourney];
Grid[Characters[#] & /@ fullRows, Frame → All, ItemSize → {1, 1},
 Background → {None, None, {{6, 1} → Cyan, {6, 26} → Cyan, {1, 13} → Cyan,
     {26, 13} → Cyan, {1, 1} → LightOrange, {1, 26} → LightOrange,
     {26, 1} → LightOrange, {26, 26} → LightOrange, {{2, 25}, {2, 25}} → LightGray,
     {5 + 1, 12 + 1} → White, {22 + 1, 9 + 1} → Lighter@LightGray}}]
```

L	M	U	L	N	I	S	I	T	A	L	I	M	R	P	E	R	E	M	I	E	L	N	L	J	E
N	D	E	A	T	H	I	N	M	A	N	Y	S	I	Z	E	S	S	O	M	E	A	N	A	M	S
T	E	S	J	U	S	T	T	H	R	E	E	T	I	M	E	S	F	O	U	N	D	W	I	T	N
E	H	N	A	M	E	S	W	O	R	N	A	W	A	Y	M	A	N	Y	W	E	R	E	W	I	T
R	T	H	M	E	S	S	A	G	E	S	O	F	L	O	V	E	C	O	B	W	E	B	S	T	N
H	H	E	M	A	M	O	N	G	T	H	E	M	F	L	O	W	E	R	S	R	U	S	T	L	S
I	E	D	I	N	A	W	I	N	D	B	R	O	U	G	H	T	A	S	I	N	G	L	E	R	V
O	O	S	E	T	O	O	N	F	A	T	H	E	R	S	R	E	S	T	I	N	G	P	L	A	L
A	C	E	A	S	F	R	O	M	S	T	O	N	E	L	I	Z	A	R	D	S	T	O	S	T	E
N	O	N	E	R	E	A	D	I	N	G	N	A	M	E	S	M	E	S	S	A	G	E	S	F	I
R	R	I	G	H	T	E	N	E	D	L	O	S	S	O	F	S	T	I	L	L	A	N	U	N	T
V	H	E	A	L	E	D	K	N	U	T	S	E	N	J	U	N	E	M	O	U	N	D	F	E	E
N	L	T	T	H	A	T	E	A	C	H	S	T	E	P	T	O	O	K	T	O	T	O	H	E	O
T	R	S	E	L	F	W	H	E	N	S	M	A	L	L	E	A	C	H	S	T	E	P	M	A	O
E	N	Y	S	T	E	P	S	T	A	K	E	N	I	N	Y	E	A	R	S	G	O	N	E	B	L
T	Y	W	H	E	N	F	O	U	N	D	F	A	T	H	E	R	S	G	R	A	V	E	H	E	H
T	A	R	T	L	E	A	P	T	U	P	I	N	A	S	I	F	C	L	I	M	B	E	D	T	M
K	O	O	F	A	M	O	U	N	T	A	I	N	I	T	A	S	T	O	N	E	L	I	K	E	I
O	A	N	A	R	C	H	B	E	N	T	T	O	R	O	S	E	U	P	O	N	G	R	A	V	Y
W	E	A	T	O	N	C	E	A	L	I	V	E	W	I	T	H	A	N	W	A	V	E	O	F	O
N	E	M	O	T	I	O	N	F	E	E	L	I	N	G	O	F	R	E	L	E	A	S	E	D	E
O	K	N	E	W	T	R	E	A	S	U	R	E	N	E	A	R	L	O	O	K	I	N	G	U	A
R	P	A	S	T	A	L	L	I	O	N	H	E	R	U	N	N	I	N	G	O	N	O	T	H	N
H	E	R	S	I	D	E	O	F	T	R	E	E	S	A	H	E	A	R	D	S	N	O	R	T	E
N	I	N	G	O	F	E	C	H	O	O	F	T	O	T	O	C	O	U	L	D	M	O	V	E	S
I	O	P	O	S	B	O	O	T	U	A	I	E	H	O	N	M	R	S	E	A	E	I	B	A	N

14	15	16

These three sentences appear to describe a specific location on the grid: a **stallion** on the other side of the **trees**, near **Treasure**. This matches the word stallion in the third line from the bottom of the gray

grid. Then, on page 22, we have the following sentences:
“Only **one** thing was certain. She had seen a horse. But was it Treasure? Or the **ghost of a memory**? ” and “Could he [the stallion] be waiting for her? Would they ride together ever again?”.

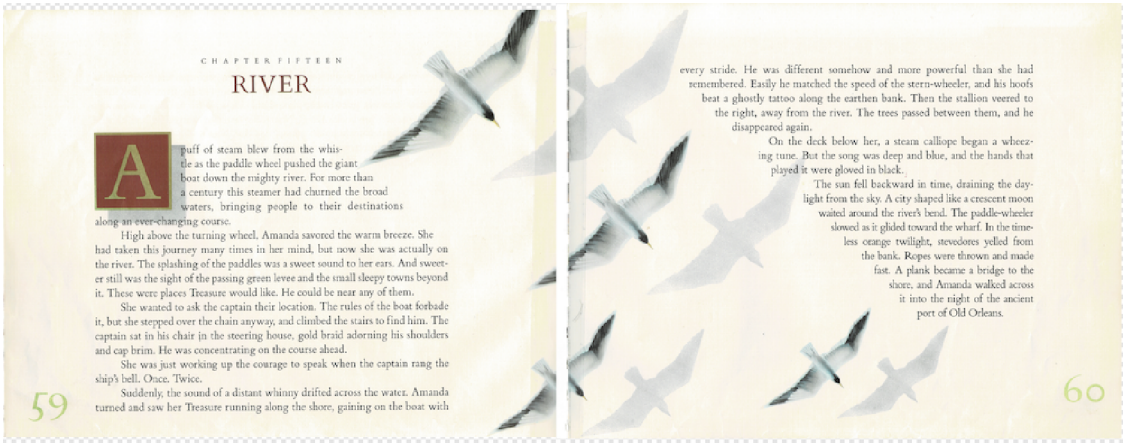
Computer **memory** is about binary numbers 0 and 1. We have a 1 (real) and we have 0 (ghost) . So the **IO** in the word stallion may be viewed as digits 0 and 1. The O (shown in a lighter gray on the map) will be our zero, starting position, when time comes to move on the cemetery map.

Now, the NILE corners tell us to go to RIVER.

RIVER

In RIVER, the Man with Black Gloves plays the music. This will guide the 4 possible directions of movement on our cemetery map. How do we know? There are two powerful clues to consider.

- The words found in the four corners of the cemetery map we constructed: **MAN DEATH MOVE IN GO**. We already know that the secret of GO in KITES is E-major with its four sharps: **C D F G**. So these four letters are the perfect candidates for our DEATH MOVES on the cemetery map.
- The 4 birds on pages 59-60 all have sharp beaks.



CHAPTER FIFTEEN

RIVER

A puff of steam blew from the whistle as the paddle wheel pushed the giant boat down the mighty river. For more than a century this steamer had churned the broad waters, bringing people to their destinations along an ever-changing course.

High above the turning wheel, Amanda savored the warm breeze. She had taken this journey many times in her mind, but now she was actually on the river. The splashing of the paddles was a sweet sound to her ears. And sweeter still was the sight of the passing green levee and the small sleepy towns beyond it. These were places Treasure would like. He could be near any of them.

She wanted to ask the captain their location. The rules of the boat forbade it, but she stepped over the chain anyway, and climbed the stairs to find him. The captain sat in his chair in the steering house, gold braid adorning his shoulders and cap brim. He was concentrating on the course ahead.

She was just working up the courage to speak when the captain rang the ship's bell. Once. Twice.

Suddenly, the sound of a distant whinny drifted across the water. Amanda turned and saw her Treasure running along the shore, gaining on the boat with

59

every stride. He was different somehow and more powerful than she had remembered. Easily he matched the speed of the stern-wheeler, and his hoofs beat a ghostly tattoo along the earthen bank. Then the stallion veered to the right, away from the river. The trees passed between them, and he disappeared again.

On the deck below her, a steam calliope began a wheezing tune. But the song was deep and blue, and the hands that played it were glowed in black.

The sun fell backward in time, draining the daylight from the sky. A city shaped like a crescent moon waited around the river's bend. The paddle-wheeler slowed as it glided toward the wharf. In the timeless orange twilight, stevedores yelled from the bank. Ropes were thrown and made fast. A plank became a bridge to the shore, and Amanda walked across it into the night of the ancient port of Old Orleans.

60

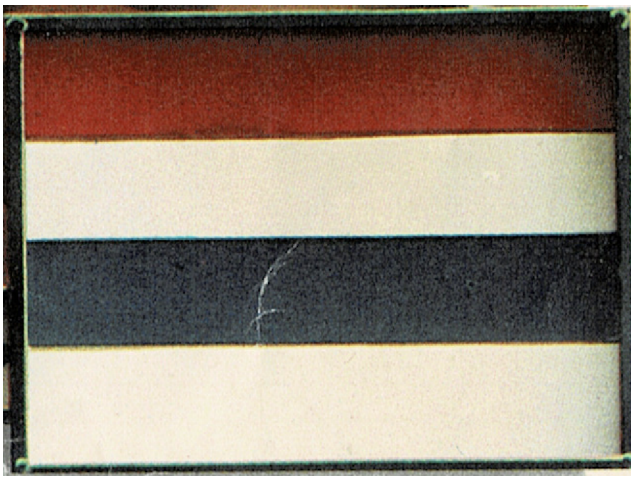
Here, it is easy to miss a crucial clue: there are **four shadow birds**. The shadow birds are not colored like the others. They are a **flat** light gray. And a key signature with 4 flats is **A-flat**, with the following flat notes: **B E A D**.
All sentences touched by shadows (including the whole first paragraph touched by the shadow of the big letter A) will provide directions using the letters **B E A D** instead of **C D F G**.

The next task is to assign the proper direction (up, right, down, left) to each of these letters, so we know how to move on our cemetery map. This requires solving the flags puzzle on pages 61-62. Here are the four flags:

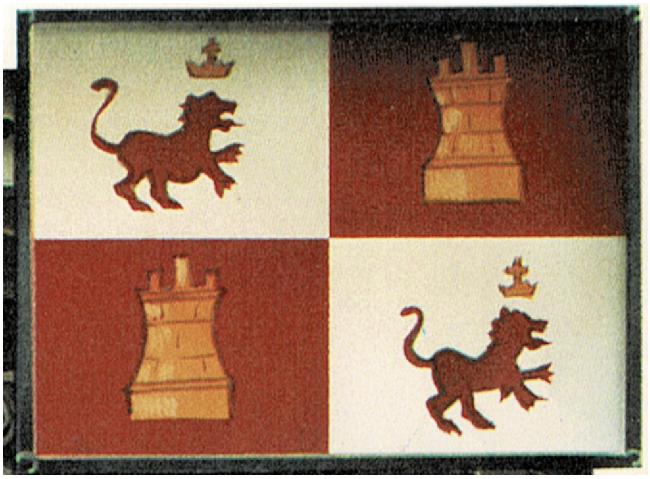
FLAG #1

FLAG #2

FLAG #3



FLAG #4



Flag #1 provides us with a text to work with. Note also the **four sharp** beaks. We need to simulate what would happen if we were moving on our cemetery grid map using instructions provided by our letters **C D F G**.
The text is made up of 25 letters: **UNION JUSTICE AND CONFIDENCE**. So we need a 5x5 grid.

```
Grid[Partition[Characters@"UNIONJUSTICEANDCONFIDENCE", 5],
  Frame → All, ItemSize → {1, 1}]
```

U	N	I	O	N
J	U	S	T	I
C	E	A	N	D
C	O	N	F	I
D	E	N	C	E

The text tells us that "the song was deep and **blue**". Flag # 3 has a third **row** that is blue. Flag #2 has 3 fleurs-de-lis on a blue background. This ancient flag was used by French **columns**. This instructs us to split our flag with row-column {3,3}, dividing it in four quarters, to match Flag #4.

```
Grid[Partition[Characters@"UNIONJUSTICEANDCONFIDENCE", 5],
  Frame → All, ItemSize → {1, 1},
  Background → {3 → Gray, 3 → Gray, {{{1, 2}, {4, 5}} → Pink, {{4, 5}, {1, 2}} → Red,
      {{1, 2}, {1, 2}} → Yellow, {{4, 5}, {4, 5}} → LightYellow}}]
```

U	N	I	O	N
J	U	S	T	I
C	E	A	N	D
C	O	N	F	I
D	E	N	C	E

Here we have colored the diagonals in different shades of yellow and red, to match the image of flag #4. One lion is fatter than the other (darker yellow) and one tower in fatter than the other (darker red). But before we go further, note that the two towers are “protected”: the dark red background spells

CODE. So the tower letters have been coded. But we know the key: **E**-major. The letter E also appears sideways on the top of the towers in flag #4. Here is the plain alphabet along side the alphabet of E.

```
Grid[{Prepend[Alphabet[], "Plain alphabet"],
   Prepend[RotateLeft[Alphabet[], 4], "Key E "]},
  Frame → All, Background → {1 → LightGray}]
```

Plain alphabet	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Key E	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d

The 8 letters behind the towers (red and pink) may now be decoded, using the above key.

```
Grid[Partition[Characters@"UNIKJJUSPECEANDYKNFIZANCE", 5],
  Frame → All, ItemSize → {1, 1},
  Background → {3 → Gray, 3 → Gray, {{1, 2}, {4, 5}} → Pink, {{4, 5}, {1, 2}} → Red,
     {{1, 2}, {1, 2}} → Yellow, {{4, 5}, {4, 5}} → LightYellow}}]
```

U	N	I	K	J
J	U	S	P	E
C	E	A	N	D
Y	K	N	F	I
Z	A	N	C	E

After decoding, the red quarter is now the heavier quarter, based on adding letter values. It was not before.
Now that things are in place, let's see what we have in terms of letters **C D F G** in the original flag text.

```
KeySelect[LetterCounts["UNIONJUSTICEANDCONFIDENCE"],
  MemberQ[Characters@"CDFG", #] &]
```

```
⟨| C → 3, D → 2, F → 1 |⟩
```

The puzzle becomes: How do I move from a starting point to an end point on the grid by moving:
3 steps in one direction
2 steps in another direction, different from the first
1 step yet in another direction, different from the two previous directions
The three chosen directions may then be matched to the corresponding letter (C,D or F). The direction not used will be matched to the letter not appearing in the text: G.

Solving this puzzle will also solve the case where "BEAD" is used, since these letters appear in corresponding numbers (0,1,2,3)

```
KeySelect[LetterCounts["UNIONJUSTICEANDCONFIDENCE"],
 MemberQ[Characters@"BEAD", #] &]
```

```
⟨| E → 3, D → 2, A → 1 |⟩
```

The question now is: What are the start and end points on the grid?
Strangely, the answer is in MEMORY, where flag and paddle-wheeler are discussed. Here are some sentences from this chapter's last paragraph:

> "Here were all the **playthings** she had ever known"
> "And in a far corner, at the back of a shelf, she
> found a wooden paddle–wheeler with a waving American **flag**"
> "...**and it felt good to have it in her hands again**."
> "For this was the boat on which she had taken
> his **first imaginary journey** in search of her missing Treasure"

The **first imaginary journey** give us the end points. The first letters of the two words "**i**maginary **j**ourney" is our path: **I to J**. Note that **I** is also used in mathematics to represent imaginary numbers. There is only one solution:

Start-I at position {4,5}

Move 3-down brings you to **E** at position {2,5} (Columns and rows are circular. When you get to one end, you continue at the other end)
Move 2-right brings you to **U** at position {2,2}
Move 1-left brings you to **J** at position {2,1}

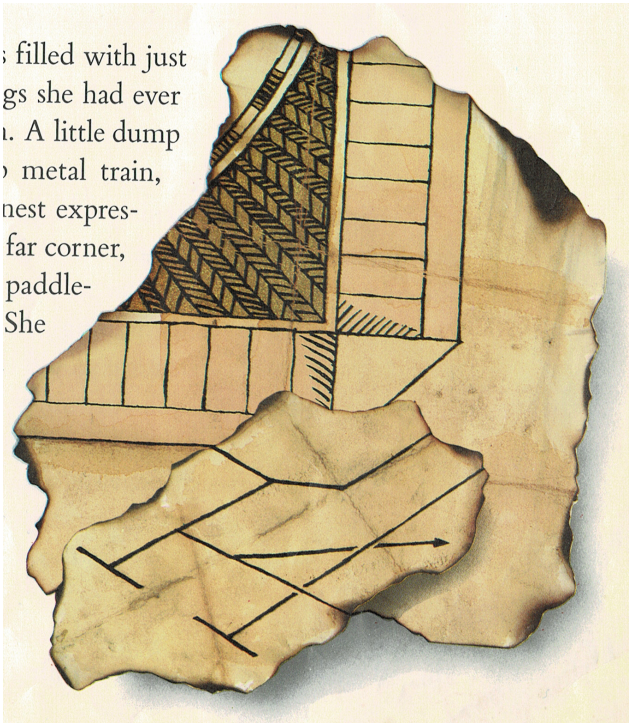
The three letters selected spell **JEU**, which is French for toy or plaything.

From these results it would appear that: C moves down. D moves right and F moves left. But there is another step. In the third sentence **felt - good** translates to **LEFT-RIGHT** (the first is an anagram, the second a synonym), and we need to **have it in her hands again**. Hands are mirror images of each other, which means right becomes left and left becomes right. Applying this change means that D moves left and F moves right. This table summarizes these findings.

```
Grid[{{"", "E-major", "A-flat"}, {"UP", "G", "B"},
   {"DOWN", "C", "E"}, {"RIGHT", "F", "A"}, {"LEFT", "D", "D"}},
  Frame → All, Background → {1 → LightGray, 1 → LightGray}]
```

	E-major	A-flat
UP	G	B
DOWN	C	E
RIGHT	F	A
LEFT	D	D

Now there is another easily overlooked clue. The paragraph in MEMORY where we find the above framed sentences is largely indented with two pieces of maps **overlapping** each other (the only instance of such an overlap). As seen in the above table, the letter **D**, indicating the Left direction, is common to both E-major and A-flat. Another **overlap**. Is there a link?



It turns out that all occurrences of the letter **D** is to be **counted twice** whether it appears in E-major sentences or A-flat-sentences.

We may now tackle the text of RIVER.

```
riverRaw =
  "A puff of steam blew from the whistle as the paddle wheel pushed the giant
    boat down the mighty river. For more than a century this steamer
    had churned the broad waters, bringing people to their destinations
    along an ever-changing course. High above the turning wheel, Amanda
    savored the warm breeze. She had taken this journey many times in
    her mind, but now she was actually on the river. The splashing of
    the paddles was a sweet sound to her ears. And sweeter still was the
    sight of the passing green levee and the small sleepy towns beyond
    it. These were places Treasure would like. He could be near any of
    them. She wanted to ask the captain their location. The rules of
    the boat forbade it, but she stepped over the chain anyway, and
    climbed the chair to find him. The captain sat in his chair in the
    steering house, gold braid adorning his shoulders and cap brim. He
    was concentrating on the course ahead. She was just working up the
    courage to speak when the captain rang the ship's bell. Once. Twice.
    Suddenly, the sound of a distant whinny drifted across the water.
    Amanda turned and saw her Treasure running along the shore, gaining
    on the boat with every stride. He was different somehow and more
    powerful than she had remembered. Easily he matched the speed of the
    stern-wheeler, and his hoofs beat a ghostly tatoo along the earthen
    bank. Then the stallion veered to the right, away from the river.
    The trees passed between them, and he disappeared again. On the deck
    below, a steam calliope began a wheezing tune. But the song was deep
    and blue, and the hands that played it were gloved in black. The
    sun fell backward in time, draining the daylight from the sky. A
    city shaped like a crescent moon waited around the river's bend. The
    paddle-wheeler slowed as it glided toward the wharf. In the timeless
    orange twilight, stevedores yelled from the bank. Ropes were thrown
    and made fast. A plank became a bridge to the shore, and Amanda
    walked across it into the night of the ancient port of Old Orleans.";
```

```
riverSentencesRaw = AssociationThread[
   Range[Length[TextSentences[riverRaw]]], TextSentences[riverRaw]];
Column@Normal@riverSentencesRaw
```

```
1 → A puff of steam blew from the whistle as the
   paddle wheel pushed the giant boat down the mighty river.
2 → For more than a century this steamer had churned the broad waters,
   bringing people to their destinations along an ever-changing course.
3 → High above the turning wheel, Amanda savored the warm breeze.
4 → She had taken this journey many
   times in her mind, but now she was actually on the river.
5 → The splashing of the paddles was a sweet sound to her ears.
6 → And sweeter still was the sight of the
   passing green levee and the small sleepy towns beyond it.
7 → These were places Treasure would like.
8 → He could be near any of them.
9 → She wanted to ask the captain their location.
10 → The rules of the boat forbade it, but she stepped
   over the chain anyway, and climbed the chair to find him.
11 → The captain sat in his chair in the steering
   house, gold braid adorning his shoulders and cap brim.
12 → He was concentrating on the course ahead.
13 → She was just working up the
   courage to speak when the captain rang the ship's bell.
14 → Once.
15 → Twice.
16 → Suddenly, the sound of a distant whinny drifted across the water.
17 → Amanda turned and saw her Treasure running
   along the shore, gaining on the boat with every stride.
18 → He was different somehow and more powerful than she had remembered.
19 → Easily he matched the speed of the stern-wheeler,
   and his hoofs beat a ghostly tatoo along the earthen bank.
20 → Then the stallion veered to the right, away from the river.
21 → The trees passed between them, and he disappeared again.
22 → On the deck below, a steam calliope began a wheezing tune.
23 → But the song was deep and blue,
   and the hands that played it were gloved in black.
24 → The sun fell backward in time, draining the daylight from the sky.
25 → A city shaped like a crescent moon waited around the river's bend.
26 → The paddle-wheeler slowed as it glided toward the wharf.
27 → In the timeless orange twilight, stevedores yelled from the bank.
28 → Ropes were thrown and made fast.
29 → A plank became a bridge to the shore, and Amanda walked
   across it into the night of the ancient port of Old Orleans.
```

We need to replace sentence #10 by a sentence in the RULES (on page 81) that specifically forbids something: “No substitution of prize is permitted”. Here is the new list of sentences.

```
riverSentences = riverSentencesRaw;
riverSentences[[Key[10]]] = "No substitution of prize is permitted.";
Column@Normal@riverSentences
```

```
1 → A puff of steam blew from the whistle as the
    paddle wheel pushed the giant boat down the mighty river.
2 → For more than a century this steamer had churned the broad waters,
    bringing people to their destinations along an ever-changing course.
3 → High above the turning wheel, Amanda savored the warm breeze.
4 → She had taken this journey many
    times in her mind, but now she was actually on the river.
5 → The splashing of the paddles was a sweet sound to her ears.
6 → And sweeter still was the sight of the
    passing green levee and the small sleepy towns beyond it.
7 → These were places Treasure would like.
8 → He could be near any of them.
9 → She wanted to ask the captain their location.
10 → No substitution of prize is permitted.
11 → The captain sat in his chair in the steering
    house, gold braid adorning his shoulders and cap brim.
12 → He was concentrating on the course ahead.
13 → She was just working up the
    courage to speak when the captain rang the ship's bell.
14 → Once.
15 → Twice.
16 → Suddenly, the sound of a distant whinny drifted across the water.
17 → Amanda turned and saw her Treasure running
    along the shore, gaining on the boat with every stride.
18 → He was different somehow and more powerful than she had remembered.
19 → Easily he matched the speed of the stern-wheeler,
    and his hoofs beat a ghostly tatoo along the earthen bank.
20 → Then the stallion veered to the right, away from the river.
21 → The trees passed between them, and he disappeared again.
22 → On the deck below, a steam calliope began a wheezing tune.
23 → But the song was deep and blue,
    and the hands that played it were gloved in black.
24 → The sun fell backward in time, draining the daylight from the sky.
25 → A city shaped like a crescent moon waited around the river's bend.
26 → The paddle-wheeler slowed as it glided toward the wharf.
27 → In the timeless orange twilight, stevedores yelled from the bank.
28 → Ropes were thrown and made fast.
29 → A plank became a bridge to the shore, and Amanda walked
    across it into the night of the ancient port of Old Orleans.
```

We now generate a set of directions, one set for each of the 29 sentences, by counting the letters **C D F G** (or **B E A D** for sentences 1, 2, 23 touched by a shadow). The letter D is counted twice.

```
directionsLetters = <||>;
(
    stcs = riverSentences[[Key[#]]];
    If[ MemberQ[{1, 2, 23}, #],
     AppendTo[directionsLetters, # → LetterCounts@
        StringDelete[ToUpperCase@stcs, Except[Characters@"BEAD"]]],
     AppendTo[directionsLetters, # →
       LetterCounts@StringDelete[ToUpperCase@stcs, Except[Characters@"CDFG"]]]
    ];
    If[NumberQ[(Last@directionsLetters)["D"]],
     directionsLetters[[-1]][[Key["D"]]] = directionsLetters[[-1]]["D"] * 2];
   ) & /@ Range[29];
Column@Normal@directionsLetters
```

```
1 → <|E → 12, A → 6, D → 8, B → 2|>
2 → <|E → 14, A → 10, D → 8, B → 2|>
3 → <|D → 4, G → 2|>
4 → <|D → 4, C → 1|>
5 → <|D → 6, F → 1, G → 1|>
6 → <|G → 3, D → 6, F → 1|>
7 → <|D → 2, C → 1|>
8 → <|F → 1, D → 2, C → 1|>
9 → <|C → 2, D → 2|>
10 → <|D → 2, F → 1|>
11 → <|D → 10, G → 3, C → 3|>
12 → <|C → 3, D → 2, G → 1|>
13 → <|G → 3, C → 2|>
14 → <|C → 1|>
15 → <|C → 1|>
16 → <|D → 12, F → 2, C → 1|>
17 → <|G → 4, D → 8|>
18 → <|D → 8, F → 3|>
19 → <|D → 6, G → 2, F → 2, C → 1|>
20 → <|F → 1, G → 1, D → 2|>
21 → <|D → 8, G → 1|>
22 → <|G → 2, C → 2, D → 2|>
23 → <|E → 9, A → 7, D → 12, B → 3|>
24 → <|D → 6, G → 2, F → 2, C → 1|>
25 → <|D → 8, C → 3|>
26 → <|D → 12, F → 1, G → 1|>
27 → <|D → 4, G → 2, F → 1|>
28 → <|D → 4, F → 1|>
29 → <|D → 10, C → 3, F → 2, G → 2|>
```

For more clarity, we will use **U** (up) - **D** (down) - **L** (left) - **R** (right) instead of the directions letters CDFG or BEAD to get our raw direction set. Here, the letter **L** (for Left, corresponding to note D) is counted twice. Here is the corresponding set of directives.

```
directionsRaw = <||>;
(
    stcs = riverSentences[[Key[#]]];
    If[ MemberQ[{1, 2, 23}, #],
     stcs = StringReplace[StringDelete[ToUpperCase@stcs,
        Except[Characters@"BEAD"]], {"B" → "U", "E" → "D", "A" → "R", "D" → "L"}];
     AppendTo[directionsRaw, # → LetterCounts@stcs],
     stcs = StringReplace[StringDelete[ToUpperCase@stcs,
        Except[Characters@"CDFG"]], {"G" → "U", "C" → "D", "F" → "R", "D" → "L"}];
     AppendTo[directionsRaw, # → LetterCounts@stcs];
    ];
    If[NumberQ[(Last@directionsRaw)["L"]],
     directionsRaw[[-1]][[Key["L"]]] = directionsRaw[[-1]]["L"] * 2];
   ) & /@ Range[29];
Column@Normal@directionsRaw
```

```
1 → <|D → 12, R → 6, L → 8, U → 2|>
2 → <|D → 14, R → 10, L → 8, U → 2|>
3 → <|L → 4, U → 2|>
4 → <|L → 4, D → 1|>
5 → <|L → 6, R → 1, U → 1|>
6 → <|U → 3, L → 6, R → 1|>
7 → <|L → 2, D → 1|>
8 → <|R → 1, L → 2, D → 1|>
9 → <|D → 2, L → 2|>
10 → <|L → 2, R → 1|>
11 → <|L → 10, U → 3, D → 3|>
12 → <|D → 3, L → 2, U → 1|>
13 → <|U → 3, D → 2|>
14 → <|D → 1|>
15 → <|D → 1|>
16 → <|L → 12, R → 2, D → 1|>
17 → <|U → 4, L → 8|>
18 → <|L → 8, R → 3|>
19 → <|L → 6, U → 2, R → 2, D → 1|>
20 → <|R → 1, U → 1, L → 2|>
21 → <|L → 8, U → 1|>
22 → <|U → 2, D → 2, L → 2|>
23 → <|D → 9, R → 7, L → 12, U → 3|>
24 → <|L → 6, U → 2, R → 2, D → 1|>
25 → <|L → 8, D → 3|>
26 → <|L → 12, R → 1, U → 1|>
27 → <|L → 4, U → 2, R → 1|>
28 → <|L → 4, R → 1|>
29 → <|L → 10, D → 3, R → 2, U → 2|>
```

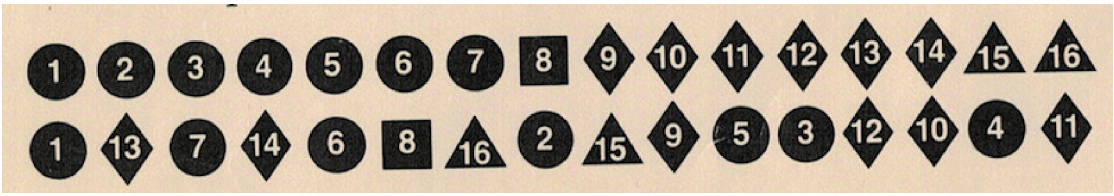
We now have a collection of 29 sets of directions. But these cannot be used directly on the cemetery map. After all, we are on a paddle-wheeler. The 29 sentences will be rotated repeatedly to specific

positions by a **driver** to select sets of directions in a specific order, suitable for use with the map. The movie clearly shows the machinery involved in rotating the wheel. This driver comes from solving the chapter CREATION.

CREATION

Solving CREATION requires the Periodic Table of Elements and the rabbit's clue card of Chapter 19. We will first talk about the clue card.

The clue card



The clue card causes a permutation of 16 items. The card shows the original sequence (line 1) and the first permutation (line 2). If the same permutation is performed many times, starting each new permutation with the result of the last permutation, you find that there are only 9 possible permutations of the original sequence, for a total of 10 lines. Doing one more permutation will bring you back to the starting sequence. We will create a function that takes 16 items (text) and compute all the permutation.

```
permute16[list_] := Module[{fullList, master, current, newList},
  i = 1;
  fullList = {Join[{i++}, list]};
  master = AssociationThread[Range[16],
    {1, 13, 7, 14, 6, 8, 16, 2, 15, 9, 5, 3, 12, 10, 4, 11}];

  current = list;
  While[Length[fullList] < 10,
   newList = Table[0, 16];
   (newList[[#]] = current[[master[[#]]]]) & /@ Range[16];
   AppendTo[fullList, Join[{i++}, newList]];
   current = newList;
  ];
  Grid[fullList, Frame → All,
   ItemSize → {1.2, 1}, Background → {1 → LightGray, None}]
 ]
```

As a demo, here we call the function with the 16 letters A through P as the initial list. (Surprisingly, the last word happens to be DIG!!).

```
permute16[Characters@"ABCDEFGHIJKLMNOP"]
```

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
2	A	M	G	N	F	H	P	B	O	I	E	C	L	J	D	K
3	A	L	P	J	H	B	K	M	D	O	F	G	C	I	N	E
4	A	C	K	I	B	M	E	L	N	D	H	P	G	O	J	F
5	A	G	E	O	M	L	F	C	J	N	B	K	P	D	I	H
6	A	P	F	D	L	C	H	G	I	J	M	E	K	N	O	B
7	A	K	H	N	C	G	B	P	O	I	L	F	E	J	D	M
8	A	E	B	J	G	P	M	K	D	O	C	H	F	I	N	L
9	A	F	M	I	P	K	L	E	N	D	G	B	H	O	J	C
10	A	H	L	O	K	E	C	F	J	N	P	M	B	D	I	G

The periodic table

Important sentences in the chapter will be framed.
Here is the most important sentence to understand CREATION:

"Seven thousand years ago, his ancestors had crossed the steppes of **As**ia to the deserts of **Ar**aby ".

As and **Ar** represent the chemical elements Arsenic and Argon. Here is the sequence of elements from As to Ar, showing their relative position in the Periodic Table of Elements.

```
Grid[{{"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "Ar"},
  {"K", "Ca", "Sc", "Ti", "V", "Cr",
   "Mn", "Fe", "Co", "Ni", "Cu", "Zn", "Ga", "Ge", "As"}
 }, Frame → All, ItemSize → {2, 1}]
```

																	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As			

Then we need to understand that **Treasure**, **Amanda** and her **father** are each represented by an element. The two letters of this element are present in their names.

- TREASURE is **Ar**. See how Ar is separated from the main row. This is the element **Ar**gon, which reminds us that **Ar** is **gon**e.
- AMANDA is **Mn**.
- FATHER is **Fe**. This symbol for Iron is indeed suitable for a blacksmith.

"But it was a gift too early given"

"Together the three of them would walk: Treasure, Amanda and her father"

Ar appears too early in the periodic table. It should be together with **Fe** and **Mn**. We need to move **Ar**. At this point we should leave the periodic table and focus on the string of letters. Note that this string is in

the reverse order compared to the order in the periodic table as we are going from **As**ia to **Ar**aby.

```
creationElementsRaw = {"As", "Ge", "Ga", "Zn", "Cu",
   "Ni", "Co", "Fe", "Mn", "Cr", "V", "Ti", "Sc", "Ca", "K", "Ar"};
Row[creationElementsRaw, "-"]
```

```
As - Ge - Ga - Zn - Cu - Ni - Co - Fe - Mn - Cr - V - Ti - Sc - Ca - K - Ar
```

"...the blacksmith would lift his daughter high onto the horse's back"

Move Treasure (Ar) with Amanda (Mn) on his back, herself supported by her father (Fe).

```
creationElementsModified = Flatten@SequenceReplace[
    DeleteCases[creationElementsRaw, "Ar"], {{"Fe", "Mn"} → {"Fe", "Mn", "Ar"}}];
Row[creationElementsModified, "-"]
```

```
As - Ge - Ga - Zn - Cu - Ni - Co - Fe - Mn - Ar - Cr - V - Ti - Sc - Ca - K
```

"...precious as the dream his daughter carried in her heart"

"That day the blacksmith began to sculpt a familiar shape from warm wax."

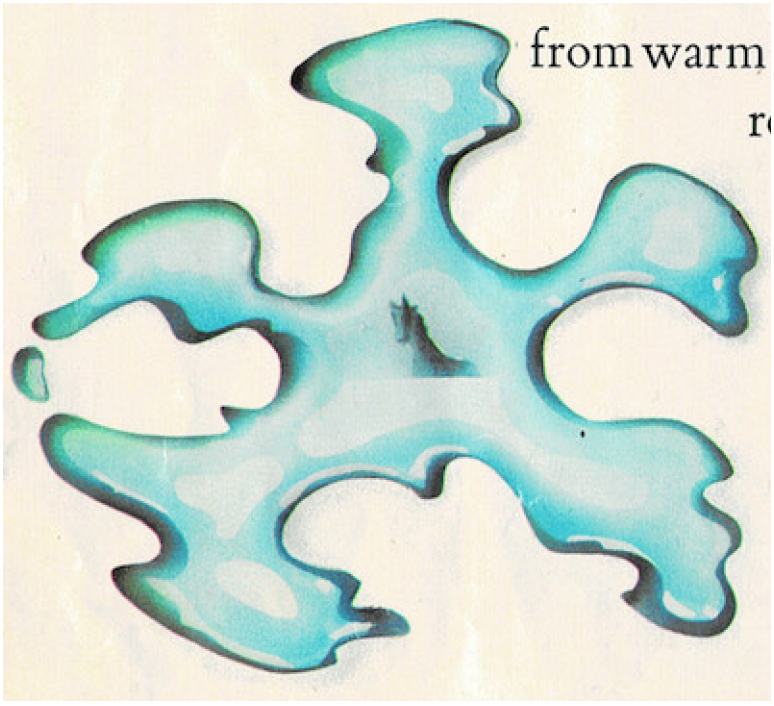
"He did it over and over..."

This text about dream and shapes and the need to do it over and over is a clue to use the clue card in chapter PARTY (a dream). As we have seen, repeating the permutation over and over sculpts a different arrangement at every step. We will call our function with the 16 chemical elements.

```
permute16[creationElementsModified]
```

1	As	Ge	Ga	Zn	Cu	Ni	Co	Fe	Mn	Ar	Cr	V	Ti	Sc	Ca	K
2	As	Ti	Co	Sc	Ni	Fe	K	Ge	Ca	Mn	Cu	Ga	V	Ar	Zn	Cr
3	As	V	K	Ar	Fe	Ge	Cr	Ti	Zn	Ca	Ni	Co	Ga	Mn	Sc	Cu
4	As	Ga	Cr	Mn	Ge	Ti	Cu	V	Sc	Zn	Fe	K	Co	Ca	Ar	Ni
5	As	Co	Cu	Ca	Ti	V	Ni	Ga	Ar	Sc	Ge	Cr	K	Zn	Mn	Fe
6	As	K	Ni	Zn	V	Ga	Fe	Co	Mn	Ar	Ti	Cu	Cr	Sc	Ca	Ge
7	As	Cr	Fe	Sc	Ga	Co	Ge	K	Ca	Mn	V	Ni	Cu	Ar	Zn	Ti
8	As	Cu	Ge	Ar	Co	K	Ti	Cr	Zn	Ca	Ga	Fe	Ni	Mn	Sc	V
9	As	Ni	Ti	Mn	K	Cr	V	Cu	Sc	Zn	Co	Ge	Fe	Ca	Ar	Ga
10	As	Fe	V	Ca	Cr	Cu	Ga	Ni	Ar	Sc	K	Ti	Ge	Zn	Mn	Co

Which one of these 9 permutations are we looking for? The one that has the shape of dreams in warm wax.



This shape is that of a **GEAR**. And the word **Ge-Ar** appears in permutation #8.

```
creationPermutation8 = {"As", "Cu", "Ge", "Ar", "Co",
   "K", "Ti", "Cr", "Zn", "Ca", "Ga", "Fe", "Ni", "Mn", "Sc", "V"};
Row[creationPermutation8, "-"]
```

```
As - Cu - Ge - Ar - Co - K - Ti - Cr - Zn - Ca - Ga - Fe - Ni - Mn - Sc - V
```

"...black sand"

"...and brushed away the clinging sand"

The **sa**nd is **As**, which clung in position 1 during all permutations. We remove it.

```
creationPermutation8NoSand = creationPermutation8[[2 ;;]];
Row[creationPermutation8NoSand, "-"]
```

```
Cu - Ge - Ar - Co - K - Ti - Cr - Zn - Ca - Ga - Fe - Ni - Mn - Sc - V
```

"...cool by the window and looked out [...] safe behind a split–rail fence, his daughter watched..."

A spit-rail is a way to mix the items by writing them in a V form (down-up-down-up...etc.) and then read the resulting text horizontally from the top row to the bottom row. To create a split-rail you must decide how many rows deep it will be. The answer is in LODGE, where Amanda safely looks out the window, through a 3x3 grid.

We will also start at the end, with the letter V. It has the shape of a split-rail and should lead, with the rest **behind**. Also, our goal is to select sentences (and sets of directions) in RIVER. This will be using the letter values of the string we are constructing. Sentence #22 in RIVER is likely the starting point: "...a steam calliope **began** a wheezing tune". The value of V is 22. Here is the split rail.

```
Grid[{{"V", "", "", "", "Fe", "", "", "", "Cr", "", "", "", "Ar"},
  {"", "Sc", "", "Ni", "", "Ga", "", "Zn", "", "Ti", "", "Co", "", "Ge", ""},
  {"", "", "Mn", "", "", "", "Ca", "", "", "", "K", "", "", "", "Cu"}}]
```

```
V         Fe         Cr        Ar
  Sc    Ni    Ga    Zn    Ti   Co    Ge
     Mn         Ca         K         Cu
```

We now read the rows.

```
creationSplitRail = {"V", "Fe", "Cr", "Ar", "Sc",
    "Ni", "Ga", "Zn", "Ti", "Co", "Ge", "Mn", "Ca", "K", "Cu"};
Row[creationSplitRail, "-"]
```

```
V – Fe – Cr – Ar – Sc – Ni – Ga – Zn – Ti – Co – Ge – Mn – Ca – K – Cu
```

"...a tiny prancing leg..."

"The moment where she would first set eyes on the golden horse"

The word **tiny** refers to elements **Ti** and **Ni**. In particular, the **eyes (I)** must be set on gold. When the clue card was given to Amanda the rabbit's warned that "**I** will mask it". Gold symbol is **Au**. So the first letter **I** is replaced by **A** the second one by **U** to form **Na** and **Tu** (or the fish TUNA).

```
creationAu = ReplaceAll[creationSplitRail, {"Ni" → "Na", "Ti" → "Tu"}];
Row[creationAu, "-"]
```

```
V – Fe – Cr – Ar – Sc – Na – Ga – Zn – Tu – Co – Ge – Mn – Ca – K – Cu
```

We now convert this to a list of 28 letters. This is the final driver.

```
creationDriver = Characters@ToUpperCase@StringJoin[creationAu]
```

```
{V, F, E, C, R, A, R, S, C, N, A, G, A, Z, N, T, U, C, O, G, E, M, N, C, A, K, C, U}
```

We have used the metaphor of the driver turning the wheel in RIVER, but a more accurate description is that we will use the value of these letters, in order, to count sentences in RIVER. The 29 sentences form a wheel. When you get to an end (sentences #1 or #29), you continue counting from the other end. At each step the directions (up, down, right, left) derived earlier for the selected sentences are retrieved to build the final, ordered set of directions.

Getting this done correctly is not immediately obvious. We will illustrate the problem with a function that counts the number of times a particular sentence is selected, assuming that we start at the top of RIVER and use the driver's letters to move, always in the same direction (top to bottom of chapter).

```
countVisits[driver_] := Module[{current, visited},
  current = 0;
  visited = {};
  (sum = current + LetterNumber[#];
     If[sum > 29, sum = sum - 29];
     AppendTo[visited, sum];
     current = sum;
    ) & /@ driver;
  KeySort@Counts[visited]

 ]
```

```
countVisits[creationDriver]
```

```
⟨| 1 → 1, 2 → 1, 4 → 2, 5 → 2, 7 → 1, 8 → 1, 10 → 2, 13 → 2, 14 → 1,
 15 → 1, 21 → 1, 22 → 2, 23 → 1, 24 → 1, 25 → 2, 26 → 2, 27 → 1, 28 → 4 |⟩
```

Our focus is on sentence #14 (“Once”) and #15 (“Twice”) in RIVER. These sentences are clues to indicate the number of times these sentences should be selected. We see above that both sentence #14 and #15 were visited once. Let’s see if running the driver twice in a row will get us what we want.

```
countVisits[Join[creationDriver, creationDriver]]
```

```
⟨| 1 → 3, 2 → 3, 3 → 1, 4 → 6, 5 → 2, 6 → 1, 7 → 2, 8 → 1,
 9 → 2, 10 → 4, 12 → 1, 13 → 3, 14 → 1, 15 → 3, 18 → 2, 19 → 1, 20 → 1,
 21 → 1, 22 → 2, 23 → 1, 24 → 1, 25 → 2, 26 → 3, 27 → 3, 28 → 5, 29 → 1 |⟩
```

Now sentence #14 was still selected only once, but sentence #15 was selected three times. So this approach does not work. Something is wrong.

Then we notice something peculiar. The first 5 letters of our driver are VFECR. Four of these letters (VECR) are on the plate of the car in ROAD. The car has different plates on the front and back. VECR in the back and MLGGS in the front. The following function takes a string and adds the letter values of these two plates. It then reduces the sum to a single letter by starting over at the beginning of the alphabet (A=27, B=28...).

```
fromSumLetters[string_] := Module[{t},
  t = Total@(LetterNumber[#] & /@ Characters@string);
  While[t > 26, t = t - 26];
  ToUpperCase@FromLetterNumber[t]
 ]
```

```
fromSumLetters["VECR"]
```

```
V
```

```
fromSumLetters["MLGGS"]
```

```
F
```

Here we have it. The letter **V** in our driver signals the forward direction (so we get to sentence #22 to begin), whereas the letter **F** signals a change to the reverse direction of counting sentences. The direction remains unchanged until either of these letters changes it. We will now modify our function accordingly.

```
countVisits2[driver_] := Module[{current, visited, func},
  current = 0;
  visited = {};
  (
     If[# == "V", func = Plus];
     If[# == "F", func = Subtract];
     sum = func[current, LetterNumber[#]];
     If[sum > 29, sum = sum - 29];
     If[sum < 1, sum = 29 + sum];
     AppendTo[visited, sum];
     current = sum;
    ) & /@ driver;
  KeySort@Counts[visited]
 ]
```

```
countVisits2[creationDriver]
```

```
⟨| 1 → 1, 2 → 2, 5 → 2, 7 → 1, 8 → 1, 10 → 2, 11 → 2, 13 → 1, 14 → 1,
 16 → 4, 17 → 1, 18 → 2, 19 → 2, 20 → 1, 21 → 1, 22 → 2, 23 → 1, 29 → 1 |⟩
```

Running the driver once gets us one visit for sentence #14 and none for #15. Let's try running the driver twice.

```
countVisits2[Join[creationDriver, creationDriver]]
```

```
⟨| 1 → 2, 2 → 3, 3 → 2, 4 → 1, 5 → 2, 7 → 1, 8 → 1, 10 → 3,
 11 → 3, 12 → 2, 13 → 1, 14 → 1, 15 → 2, 16 → 4, 17 → 2, 18 → 3, 19 → 2,
 20 → 3, 21 → 3, 22 → 2, 23 → 2, 24 → 1, 26 → 4, 27 → 1, 28 → 2, 29 → 3 |⟩
```

This is what we want. Sentence #14 once, and #15 twice.
There is a third verification sentence: “Easily he matched the speed of the stern-wheeler, and his hoofs beat a ghostly tatoo along the earthen bank”.
This is sentence #19, which has been visited twice, matching the **hoofs beat** of the pig running along the bank on page 61 (two feet on the ground and two above ground).

The following function takes the list of letters derived in CREATION and uses it with the list of raw directions derived in RIVER. The net result will be an ordered list of directions suitable for use with the cemetery map.

```
getFinalDirections[driver_, rawDirections_] := Module[{current, visited, func},
  current = 0;
  visited = {};
  (
     If[# == "V", func = Plus];
     If[# == "F", func = Subtract];
     sum = func[current, LetterNumber[#]];
     If[sum > 29, sum = sum - 29];
     If[sum < 1, sum = 29 + sum];
     AppendTo[visited, sum];
     current = sum;
    ) & /@ driver;
  rawDirections[[#]] & /@ visited
 ]
```

We now call the function. Note that we already have our raw list of directions from RIVER. The driver is repeated twice to select a total of 56 sets of directions now in the correct order to use on the map.
Directions are **U** (up) - **D** (down) - **L** (left) - **R** (right) .

```
riverFinalDirections =
 getFinalDirections[Join[creationDriver, creationDriver], directionsRaw]
```

```
{⟨|U → 2, D → 2, L → 2|⟩, ⟨|L → 12, R → 2, D → 1|⟩, ⟨|L → 10, U → 3, D → 3|⟩,
 ⟨|R → 1, L → 2, D → 1|⟩, ⟨|L → 6, U → 2, R → 2, D → 1|⟩, ⟨|L → 8, R → 3|⟩,
 ⟨|L → 10, D → 3, R → 2, U → 2|⟩, ⟨|L → 2, R → 1|⟩, ⟨|L → 2, D → 1|⟩, ⟨|U → 2, D → 2, L → 2|⟩,
 ⟨|L → 8, U → 1|⟩, ⟨|D → 1|⟩, ⟨|U → 3, D → 2|⟩, ⟨|L → 12, R → 2, D → 1|⟩,
 ⟨|D → 14, R → 10, L → 8, U → 2|⟩, ⟨|L → 10, U → 3, D → 3|⟩, ⟨|L → 6, U → 2, R → 2, D → 1|⟩,
 ⟨|L → 12, R → 2, D → 1|⟩, ⟨|D → 12, R → 6, L → 8, U → 2|⟩, ⟨|D → 9, R → 7, L → 12, U → 3|⟩,
 ⟨|L → 8, R → 3|⟩, ⟨|L → 6, R → 1, U → 1|⟩, ⟨|R → 1, U → 1, L → 2|⟩, ⟨|U → 4, L → 8|⟩,
 ⟨|L → 12, R → 2, D → 1|⟩, ⟨|L → 6, R → 1, U → 1|⟩, ⟨|D → 14, R → 10, L → 8, U → 2|⟩,
 ⟨|L → 2, R → 1|⟩, ⟨|L → 4, U → 2|⟩, ⟨|L → 12, R → 1, U → 1|⟩, ⟨|L → 8, U → 1|⟩,
 ⟨|L → 8, R → 3|⟩, ⟨|L → 10, D → 3, R → 2, U → 2|⟩, ⟨|L → 4, R → 1|⟩, ⟨|L → 2, R → 1|⟩,
 ⟨|R → 1, U → 1, L → 2|⟩, ⟨|U → 4, L → 8|⟩, ⟨|L → 4, U → 2|⟩, ⟨|D → 14, R → 10, L → 8, U → 2|⟩,
 ⟨|L → 6, U → 2, R → 2, D → 1|⟩, ⟨|D → 9, R → 7, L → 12, U → 3|⟩, ⟨|L → 12, R → 1, U → 1|⟩,
 ⟨|D → 3, L → 2, U → 1|⟩, ⟨|L → 8, U → 1|⟩, ⟨|L → 10, D → 3, R → 2, U → 2|⟩,
 ⟨|L → 12, R → 1, U → 1|⟩, ⟨|L → 10, U → 3, D → 3|⟩, ⟨|L → 4, D → 1|⟩,
 ⟨|L → 4, R → 1|⟩, ⟨|D → 1|⟩, ⟨|D → 12, R → 6, L → 8, U → 2|⟩, ⟨|L → 4, U → 2, R → 1|⟩,
 ⟨|L → 12, R → 1, U → 1|⟩, ⟨|D → 1|⟩, ⟨|D → 3, L → 2, U → 1|⟩, ⟨|R → 1, U → 1, L → 2|⟩}
```

As usual there is a way to verify the accuracy of this result. Here is the total displacement in each direction.

```
totalDisplacementPerDirections = Merge[Join[riverFinalDirections], Total]
```

```
⟨|U → 72, D → 127, L → 362, R → 101|⟩
```

Given that U and D cancel each others, same for L and R, we end up with 127-72 = **55 Down** and 362-101 = **261 Left**

Given that the cemetery map is a 24 x 24 grid and that rows and columns are circular, the above movement is the same as **7 Down** and **21 Left**.

If you start on the **O** of the word stallion on the map (our zero position), move 7 Down and 21 Left, you end up on the **X** position on the map (occupied by the letter M). This is good.

NOTE: If you use the map provided earlier to verify this, use only the **24x24 gray area** and ignore the presence of the border with the Journey's text.

KITES

The way to use RIVER's final directions on the map is relatively straightforward. You get a starting point (we already determined that the **o** of "stallion" will be used), and you apply the directions in order, stepping on 56 letters of the cemetery grid as you go along. Unfortunately, just doing this will not get you anywhere without an understanding of the chapter KITES. Let's look at some of its sentences.

First we will look at what is said about the first kite.

"It returned to Earth...without a clue to her Treasure "

"A squall of clouds and a scissors of wind snapped
the thin string and the kite flew off...perhaps to join her horse"

So the string of letters collected is not to be kept, but the kite may still be useful as it joins her horse.

Then Amanda starts sending kites every year.

"Each year she made new kites to fly high, to escape, to find her missing stallion"

"There were big kites and small"
"Kites common and kites strange"
"Kites with fish and frogs"
"Kites with birds and the things they eat"

"Twelve springs (and twelve kites) passed without success"

So we must make 12 kites. But what exactly are these kites? The clue is in the above four sentences describing kites: **they are all made up of pairs of things**.
It turns out that when our 56 sets of directions are used to move on the map, the squares stepped on create a pattern: 52 spots are visited once and **4 spots are visited twice** (a pair). These four spots form a kite. If you run these 56 sets of directions 12 times in a row (each time starting on the position where you ended up in the last run), these 4 spots visited twice effectively produce 12 kites of different shapes and size, some very strange.

Then you run a thirteenth kite.

"...she painted a new kite and made it exactly like the first kite the storm had taken"

"...no gust of wind could break the string or take the kite away"

On kite #13 **you keep the string**, that is the 56 letters (4 of them repeated twice) produced by this last run of the 56 sets of directions. But you also want to keep the first kite (**without its string**). Here are the 13 kites (in red) and the string of the **last** kite (in yellow).

The following function takes a starting coordinate, the set of directions, and the number of runs to complete, while traveling on a 24 x 24 grid. It then collects all the coordinates visited (row, column), grouped by runs.

```
getVisitedCoordinates[startCoord_, directions_, runs_] :=
 Module[{currentCoord, row, col, visited, listVisited, set},

  listVisited = {};
  {row, col} = startCoord;
  Do[
   visited = {};
   i = 1;
   While[i ≤ Length[directions],
    set = directions[[i]];

    If[NumberQ[set["U"]], row = row - set["U"]];
    If[NumberQ[set["D"]], row = row + set["D"]];
    If[NumberQ[set["R"]], col = col + set["R"]];
    If[NumberQ[set["L"]], col = col - set["L"]];

    If[row > 24, row = row - 24];
    If[row < 1, row = 24 + row];
    If[col > 24, col = col - 24];
    If[col < 1, col = 24 + col];
    AppendTo[visited, {row, col}];
    i++;
   ];
   AppendTo[listVisited, visited];
   , 13];
  listVisited
 ]
```

We call this function with our starting coordinate corresponding to the O of “stallion”, and we ask that our final directions be run 13 times. The result is 13x56 = 728 visited coordinates. Horizontal lines separate each of the 13 runs.

```
visitedCoordinates = getVisitedCoordinates[{22, 9}, riverFinalDirections, 13];
Column[visitedCoordinates, Dividers → {False, All}]
```

```
{{22, 7}, {23, 21}, {23, 11}, {24, 10}, {23, 6}, {23, 1}, {24, 17}, {24, 16},
 {1, 14}, {1, 12}, {24, 4}, {1, 4}, {24, 4}, {1, 18}, {13, 20}, {13, 10},
 {12, 6}, {13, 20}, {23, 18}, {5, 13}, {5, 8}, {4, 3}, {3, 2}, {23, 18},
 {24, 8}, {23, 3}, {11, 5}, {11, 4}, {9, 24}, {8, 13}, {7, 5}, {7, 24},
 {8, 16}, {8, 13}, {8, 12}, {7, 11}, {3, 3}, {1, 23}, {13, 1}, {12, 21},
```

```
{{5, 10}, {6, 24}, {6, 14}, {7, 13}, {6, 9}, {6, 4}, {7, 20}, {7, 19},
 {8, 17}, {8, 15}, {7, 7}, {8, 7}, {7, 7}, {8, 21}, {20, 23}, {20, 13},
 {19, 9}, {20, 23}, {6, 21}, {12, 16}, {12, 11}, {11, 6}, {10, 5}, {6, 21},
 {7, 11}, {6, 6}, {18, 8}, {18, 7}, {16, 3}, {15, 16}, {14, 8}, {14, 3},
 {15, 19}, {15, 16}, {15, 15}, {14, 14}, {10, 6}, {8, 2}, {20, 4}, {19, 24},
 {1, 19}, {24, 8}, {2, 6}, {1, 22}, {2, 14}, {1, 3}, {1, 17}, {2, 13},
 {2, 10}, {3, 10}, {13, 8}, {11, 5}, {10, 18}, {11, 18}, {13, 16}, {12, 15}}
```

```
{{12, 13}, {13, 3}, {13, 17}, {14, 16}, {13, 12}, {13, 7}, {14, 23},
 {14, 22}, {15, 20}, {15, 18}, {14, 10}, {15, 10}, {14, 10}, {15, 24}, {3, 2},
 {3, 16}, {2, 12}, {3, 2}, {13, 24}, {19, 19}, {19, 14}, {18, 9}, {17, 8},
 {13, 24}, {14, 14}, {13, 9}, {1, 11}, {1, 10}, {23, 6}, {22, 19}, {21, 11},
 {21, 6}, {22, 22}, {22, 19}, {22, 18}, {21, 17}, {17, 9}, {15, 5}, {3, 7},
 {2, 3}, {8, 22}, {7, 11}, {9, 9}, {8, 1}, {9, 17}, {8, 6}, {8, 20}, {9, 16},
 {9, 13}, {10, 13}, {20, 11}, {18, 8}, {17, 21}, {18, 21}, {20, 19}, {19, 18}}
```

```
{{19, 16}, {20, 6}, {20, 20}, {21, 19}, {20, 15}, {20, 10}, {21, 2}, {21, 1},
 {22, 23}, {22, 21}, {21, 13}, {22, 13}, {21, 13}, {22, 3}, {10, 5}, {10, 19},
 {9, 15}, {10, 5}, {20, 3}, {2, 22}, {2, 17}, {1, 12}, {24, 11}, {20, 3},
 {21, 17}, {20, 12}, {8, 14}, {8, 13}, {6, 9}, {5, 22}, {4, 14}, {4, 9},
 {5, 1}, {5, 22}, {5, 21}, {4, 20}, {24, 12}, {22, 8}, {10, 10}, {9, 6},
 {15, 1}, {14, 14}, {16, 12}, {15, 4}, {16, 20}, {15, 9}, {15, 23}, {16, 19},
 {16, 16}, {17, 16}, {3, 14}, {1, 11}, {24, 24}, {1, 24}, {3, 22}, {2, 21}}
```

```
{{2, 19}, {3, 9}, {3, 23}, {4, 22}, {3, 18}, {3, 13}, {4, 5}, {4, 4},
 {5, 2}, {5, 24}, {4, 16}, {5, 16}, {4, 16}, {5, 6}, {17, 8}, {17, 22},
 {16, 18}, {17, 8}, {3, 6}, {9, 1}, {9, 20}, {8, 15}, {7, 14}, {3, 6},
 {4, 20}, {3, 15}, {15, 17}, {15, 16}, {13, 12}, {12, 1}, {11, 17}, {11, 12},
 {12, 4}, {12, 1}, {12, 24}, {11, 23}, {7, 15}, {5, 11}, {17, 13}, {16, 9},
 {22, 4}, {21, 17}, {23, 15}, {22, 7}, {23, 23}, {22, 12}, {22, 2}, {23, 22},
 {23, 19}, {24, 19}, {10, 17}, {8, 14}, {7, 3}, {8, 3}, {10, 1}, {9, 24}}
```

```
{{9, 22}, {10, 12}, {10, 2}, {11, 1}, {10, 21}, {10, 16}, {11, 8}, {11, 7},
 {12, 5}, {12, 3}, {11, 19}, {12, 19}, {11, 19}, {12, 9}, {24, 11}, {24, 1},
 {23, 21}, {24, 11}, {10, 9}, {16, 4}, {16, 23}, {15, 18}, {14, 17}, {10, 9},
 {11, 23}, {10, 18}, {22, 20}, {22, 19}, {20, 15}, {19, 4}, {18, 20},
 {18, 15}, {19, 7}, {19, 4}, {19, 3}, {18, 2}, {14, 18}, {12, 14}, {24, 16},
 {23, 12}, {5, 7}, {4, 20}, {6, 18}, {5, 10}, {6, 2}, {5, 15}, {5, 5}, {6, 1},
 {6, 22}, {7, 22}, {17, 20}, {15, 17}, {14, 6}, {15, 6}, {17, 4}, {16, 3}}
```

```
{{16, 1}, {17, 15}, {17, 5}, {18, 4}, {17, 24}, {17, 19}, {18, 11}, {18, 10},
 {19, 8}, {19, 6}, {18, 22}, {19, 22}, {18, 22}, {19, 12}, {7, 14}, {7, 4},
 {6, 24}, {7, 14}, {17, 12}, {23, 7}, {23, 2}, {22, 21}, {21, 20}, {17, 12},
 {18, 2}, {17, 21}, {5, 23}, {5, 22}, {3, 18}, {2, 7}, {1, 23}, {1, 18},
 {2, 10}, {2, 7}, {2, 6}, {1, 5}, {21, 21}, {19, 17}, {7, 19}, {6, 15},
 {12, 10}, {11, 23}, {13, 21}, {12, 13}, {13, 5}, {12, 18}, {12, 8}, {13, 4},
 {13, 1}, {14, 1}, {24, 23}, {22, 20}, {21, 9}, {22, 9}, {24, 7}, {23, 6}}
```

```
{{23, 4}, {24, 18}, {24, 8}, {1, 7}, {24, 3}, {24, 22}, {1, 14}, {1, 13},
 {2, 11}, {2, 9}, {1, 1}, {2, 1}, {1, 1}, {2, 15}, {14, 17}, {14, 7},
 {13, 3}, {14, 17}, {24, 15}, {6, 10}, {6, 5}, {5, 24}, {4, 23}, {24, 15},
 {1, 5}, {24, 24}, {12, 2}, {12, 1}, {10, 21}, {9, 10}, {8, 2}, {8, 21},
 {9, 13}, {9, 10}, {9, 9}, {8, 8}, {4, 24}, {2, 20}, {14, 22}, {13, 18},
 {19, 13}, {18, 2}, {20, 24}, {19, 16}, {20, 8}, {19, 21}, {19, 11}, {20, 7},
 {20, 4}, {21, 4}, {7, 2}, {5, 23}, {4, 12}, {5, 12}, {7, 10}, {6, 9}}
```

```
{{6, 7}, {7, 21}, {7, 11}, {8, 10}, {7, 6}, {7, 1}, {8, 17}, {8, 16},
 {9, 14}, {9, 12}, {8, 4}, {9, 4}, {8, 4}, {9, 18}, {21, 20}, {21, 10},
 {20, 6}, {21, 20}, {7, 18}, {13, 13}, {13, 8}, {12, 3}, {11, 2}, {7, 18},
 {8, 8}, {7, 3}, {19, 5}, {19, 4}, {17, 24}, {16, 13}, {15, 5}, {15, 24},
 {16, 16}, {16, 13}, {16, 12}, {15, 11}, {11, 3}, {9, 23}, {21, 1}, {20, 21},
 {2, 16}, {1, 5}, {3, 3}, {2, 19}, {3, 11}, {2, 24}, {2, 14}, {3, 10},
 {3, 7}, {4, 7}, {14, 5}, {12, 2}, {11, 15}, {12, 15}, {14, 13}, {13, 12}}
```

```
{{13, 10}, {14, 24}, {14, 14}, {15, 13}, {14, 9}, {14, 4}, {15, 20}, {15, 19},
 {16, 17}, {16, 15}, {15, 7}, {16, 7}, {15, 7}, {16, 21}, {4, 23}, {4, 13},
 {3, 9}, {4, 23}, {14, 21}, {20, 16}, {20, 11}, {19, 6}, {18, 5}, {14, 21},
 {15, 11}, {14, 6}, {2, 8}, {2, 7}, {24, 3}, {23, 16}, {22, 8}, {22, 3},
 {23, 19}, {23, 16}, {23, 15}, {22, 14}, {18, 6}, {16, 2}, {4, 4}, {3, 24},
 {9, 19}, {8, 8}, {10, 6}, {9, 22}, {10, 14}, {9, 3}, {9, 17}, {10, 13},
 {10, 10}, {11, 10}, {21, 8}, {19, 5}, {18, 18}, {19, 18}, {21, 16}, {20, 15}}
```

```
{{20, 13}, {21, 3}, {21, 17}, {22, 16}, {21, 12}, {21, 7}, {22, 23}, {22, 22},
 {23, 20}, {23, 18}, {22, 10}, {23, 10}, {22, 10}, {23, 24}, {11, 2},
 {11, 16}, {10, 12}, {11, 2}, {21, 24}, {3, 19}, {3, 14}, {2, 9}, {1, 8},
 {21, 24}, {22, 14}, {21, 9}, {9, 11}, {9, 10}, {7, 6}, {6, 19}, {5, 11},
 {5, 6}, {6, 22}, {6, 19}, {6, 18}, {5, 17}, {1, 9}, {23, 5}, {11, 7}, {10, 3},
 {16, 22}, {15, 11}, {17, 9}, {16, 1}, {17, 17}, {16, 6}, {16, 20}, {17, 16},
 {17, 13}, {18, 13}, {4, 11}, {2, 8}, {1, 21}, {2, 21}, {4, 19}, {3, 18}}
```

```
{{3, 16}, {4, 6}, {4, 20}, {5, 19}, {4, 15}, {4, 10}, {5, 2}, {5, 1},
 {6, 23}, {6, 21}, {5, 13}, {6, 13}, {5, 13}, {6, 3}, {18, 5}, {18, 19},
 {17, 15}, {18, 5}, {4, 3}, {10, 22}, {10, 17}, {9, 12}, {8, 11}, {4, 3},
 {5, 17}, {4, 12}, {16, 14}, {16, 13}, {14, 9}, {13, 22}, {12, 14}, {12, 9},
 {13, 1}, {13, 22}, {13, 21}, {12, 20}, {8, 12}, {6, 8}, {18, 10}, {17, 6},
 {23, 1}, {22, 14}, {24, 12}, {23, 4}, {24, 20}, {23, 9}, {23, 23}, {24, 19},
 {24, 16}, {1, 16}, {11, 14}, {9, 11}, {8, 24}, {9, 24}, {11, 22}, {10, 21}}
```

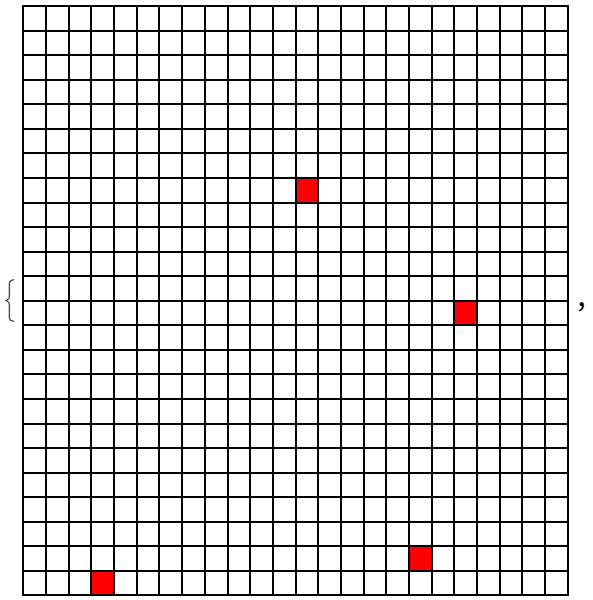
```
{{10, 19}, {11, 9}, {11, 23}, {12, 22}, {11, 18}, {11, 13}, {12, 5},
 {12, 4}, {13, 2}, {13, 24}, {12, 16}, {13, 16}, {12, 16}, {13, 6}, {1, 8},
 {1, 22}, {24, 18}, {1, 8}, {11, 6}, {17, 1}, {17, 20}, {16, 15}, {15, 14},
 {11, 6}, {12, 20}, {11, 15}, {23, 17}, {23, 16}, {21, 12}, {20, 1}, {19, 17},
 {19, 12}, {20, 4}, {20, 1}, {20, 24}, {19, 23}, {15, 15}, {13, 11}, {1, 13},
 {24, 9}, {6, 4}, {5, 17}, {7, 15}, {6, 7}, {7, 23}, {6, 12}, {6, 2}, {7, 22},
 {7, 19}, {8, 19}, {18, 17}, {16, 14}, {15, 3}, {16, 3}, {18, 1}, {17, 24}}
```

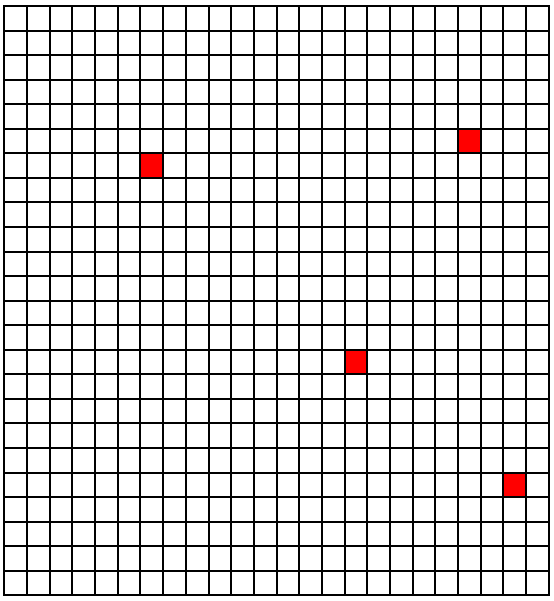
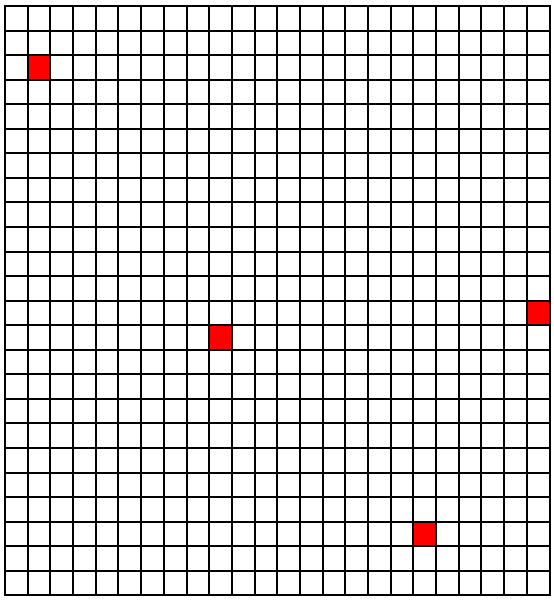
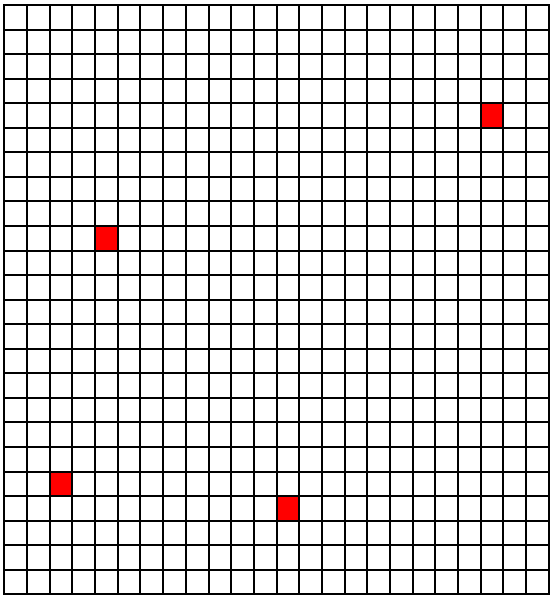
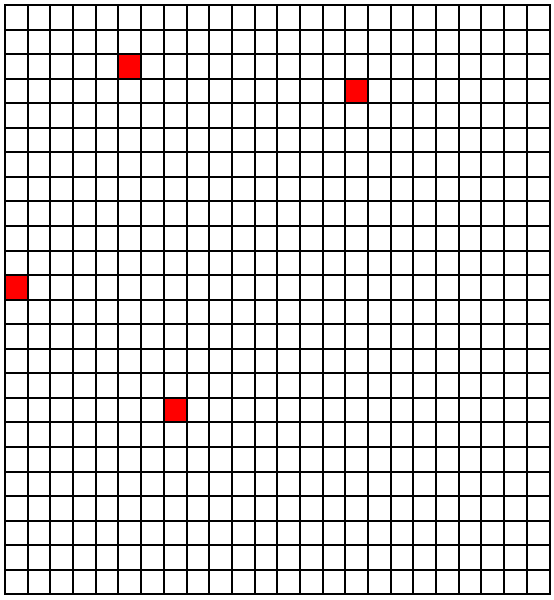
We will now create a function that uses the 4 coordinates visited twice in each runs to display the 13 kites.

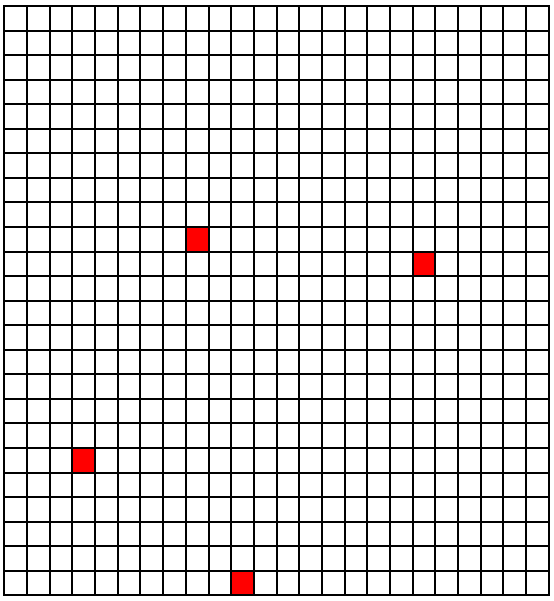
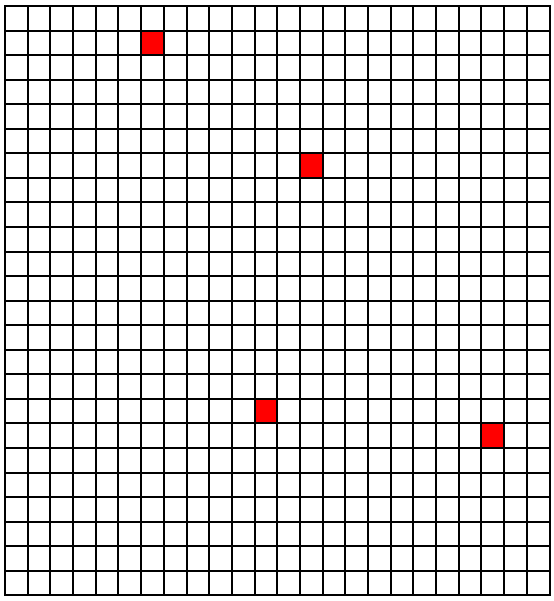
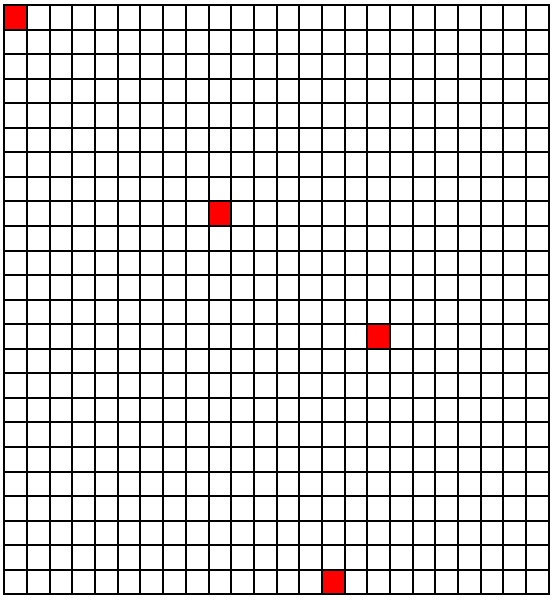
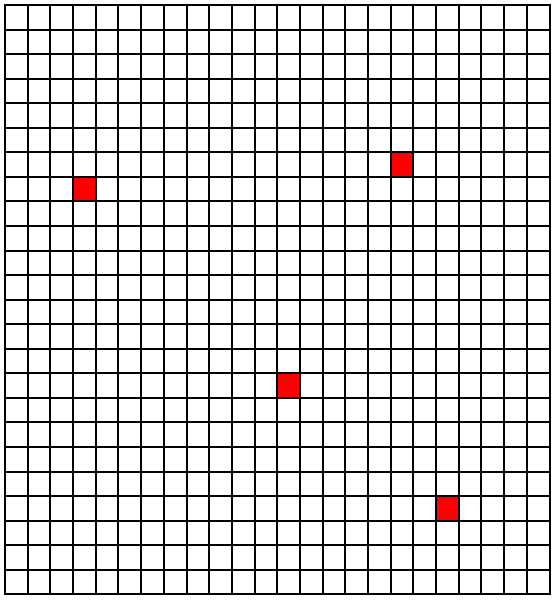
```
showKites[coordList_] := Module[{},
  result = {};
  g = Counts[#] & /@ coordList;
  i = 1;
  While[i ≤ Length[g],
   AppendTo[result, Select[g[[i]], # == 2 &]];
   i++
  ];
  Grid[Table[Table[" ", 24], 24], Frame → All, ItemSize → {0.4, 0.1},
     Background → {None, None, Thread[Rule[Keys[#], Red]]}] & /@ result
 ]
```

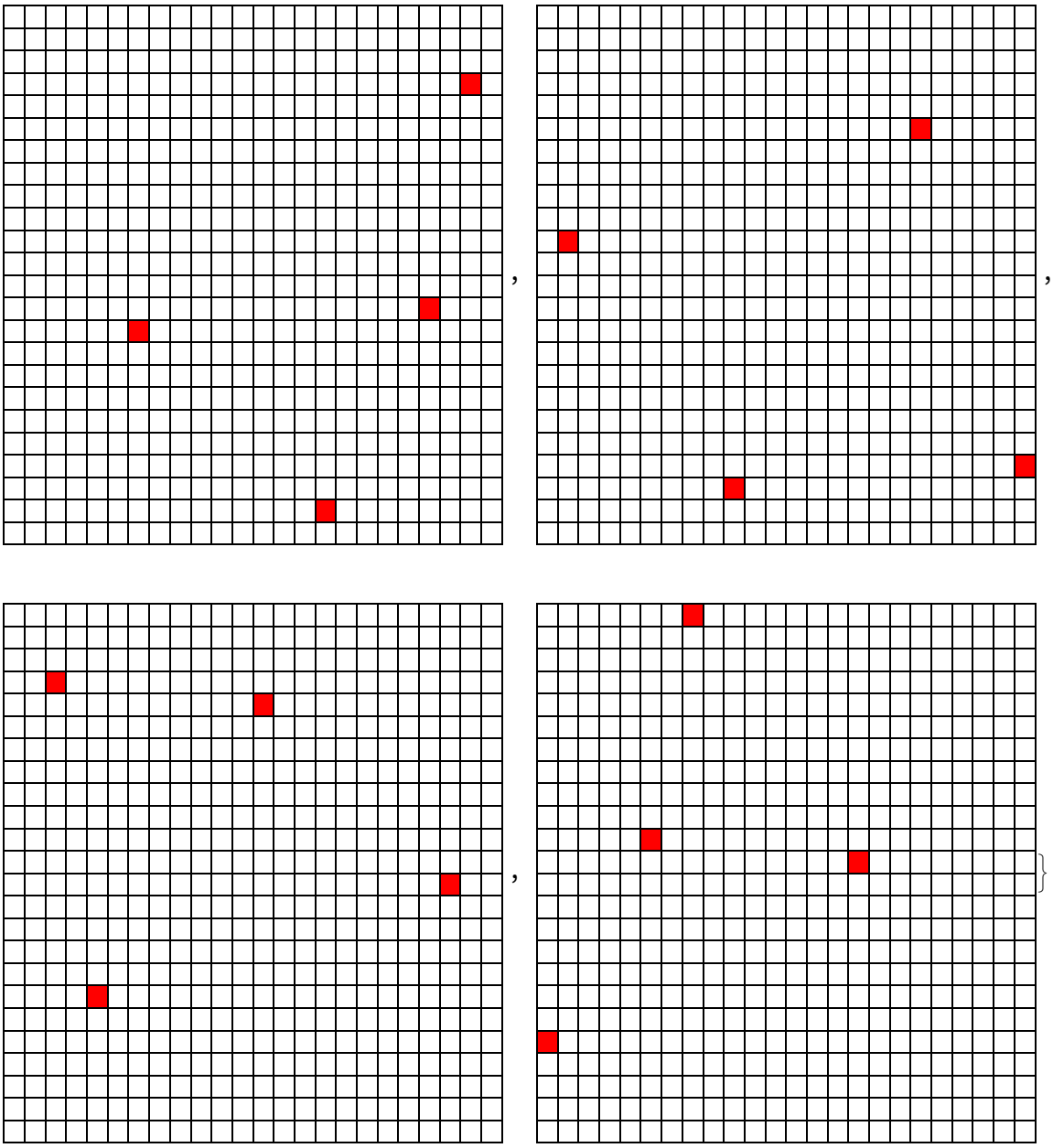
Calling this function and displaying the 13 kites on a 24x24 grid. At this point the text is not shown.

```
showKites[visitedCoordinates]
```




,

,

,

,


,

,

,

,



We can readily see that constructing a good looking kite with the 4 points is easier in some cases, but harder in others. We have kites of different shapes, some common and some strange. Which one is the strangest? If you had to you would probably choose kite #8. This is noted in CEMETERY:

"On the **eight** day of her search she found a cemetery that was older and **odder** than any she had seen before. A corona of clouds spread above it, and a **line of trees** encircled it".

Yes, this "line of **three** kites" is an issue.

As discussed, kites #2 to #12 are not used. The full string of letters visited in Kite #13 is what is important in deriving the location of the golden horse. However, the four points of kite #1 also play a role.

We now create a function to display kite #1 (in green), kite #13 (in red) and the string of locations visited in kite #13 (in light yellow), all on the same cemetery text.

```
showKites1And13[coordList_, cemeteryWords_] := Module[
  {k1, k13, k1Points, green, yellow, red, k13Points, background, cemeteryMap},
  k1 = coordList[[1]];
  k13 = coordList[[13]];
  k1Points = Keys[Select[Counts@k1, # == 2 &]];
  green = Thread[Rule[k1Points, Green]];
  yellow = Thread[Rule[k13, LightYellow]];
  k13Points = Keys[Select[Counts@k13, # == 2 &]];
  red = Thread[Rule[k13Points, Red]];
  bg = {{1, 24}, {1, 24}} → LightGray;
  background = Prepend[Join[yellow, green, red], bg];
  cemeteryMap = Partition[Characters@cemeteryWords, 24];
  Grid[cemeteryMap, Frame → All,
   ItemSize → {1, 1}, Background → {None, None, background}]
 ]
```

The function is called with our list of coordinates and our final version of the cemetery words, created from page 21 in CEMETERY.

```
k1k13 = showKites1And13[visitedCoordinates, cemeteryWordsP21Mod5]
```

D	E	A	T	H	I	N	M	A	N	Y	S	I	Z	E	S	S	O	M	E	A	N	A	M
E	S	J	U	S	T	T	H	R	E	E	T	I	M	E	S	F	O	U	N	D	W	I	T
H	N	A	M	E	S	W	O	R	N	A	W	A	Y	M	A	N	Y	W	E	R	E	W	I
T	H	M	E	S	S	A	G	E	S	O	F	L	O	V	E	C	O	B	W	E	B	S	T
H	E	M	A	M	O	N	G	T	H	E	M	F	L	O	W	E	R	S	R	U	S	T	L
E	D	I	N	A	W	I	N	D	B	R	O	U	G	H	T	A	S	I	N	G	L	E	R
O	S	E	T	O	O	N	F	A	T	H	E	R	S	R	E	S	T	I	N	G	P	L	A
C	E	A	S	F	R	O	M	S	T	O	N	E	L	I	Z	A	R	D	S	T	O	S	T
O	N	E	R	E	A	D	I	N	G	N	A	M	E	S	M	E	S	S	A	G	E	S	F
R	I	G	H	T	E	N	E	D	L	O	S	S	O	F	S	T	I	L	L	A	N	U	N
H	E	A	L	E	D	K	N	U	T	S	E	N	J	U	N	E	M	O	U	N	D	F	E
L	T	T	H	A	T	E	A	C	H	S	T	E	P	T	O	O	K	T	O	T	O	H	E
R	S	E	L	F	W	H	E	N	S	M	A	L	L	E	A	C	H	S	T	E	P	M	A
N	Y	S	T	E	P	S	T	A	K	E	N	I	N	Y	E	A	R	S	G	O	N	E	B
Y	W	H	E	N	F	O	U	N	D	F	A	T	H	E	R	S	G	R	A	V	E	H	E
A	R	T	L	E	A	P	T	U	P	I	N	A	S	I	F	C	L	I	M	B	E	D	T
O	O	F	A	M	O	U	N	T	A	I	N	I	T	A	S	T	O	N	E	L	I	K	E
A	N	A	R	C	H	B	E	N	T	T	O	R	O	S	E	U	P	O	N	G	R	A	V
E	A	T	O	N	C	E	A	L	I	V	E	W	I	T	H	A	N	W	A	V	E	O	F
E	M	O	T	I	O	N	F	E	E	L	I	N	G	O	F	R	E	L	E	A	S	E	D
K	N	E	W	T	R	E	A	S	U	R	E	N	E	A	R	L	O	O	K	I	N	G	U
P	A	S	T	A	L	L	I	O	N	H	E	R	U	N	N	I	N	G	O	N	O	T	H
E	R	S	I	D	E	O	F	T	R	E	E	S	A	H	E	A	R	D	S	N	O	R	T
I	N	G	O	F	E	C	H	O	O	F	T	O	T	O	C	O	U	L	D	M	O	V	E

This is our **master grid**.

It turns out that the role of the red and green kites is to select columns.
The four columns identified by the **green** kite (**4-13-18-20**) are involved in determining LATITUDE.
The four columns identified by the **red** kite (**1-6-8-16**) are involved in determining LONGITUDE.
Only the letters visited in these 8 columns will be used (except when directed by a specific rule). This includes the yellow letters, visited once, and the red letters, visited twice. Green letters are not part of the string of 56 letters visited in Kite #13. **These green letters only indicate the columns, but the letters themselves will not be used in the computation of latitude**.

We cannot proceed with this now as we are still missing components. It is the horses walking on the border (Journey's letters) that will **modify** the results given by letters in these columns.

Before we leave KITES, what about Aunt Margaret, discussed at length in KITES, and again in CEMETERY?
AUNT MARGARET is an anagram for **UTTER ANAGRAM**.
Are the kites uttering an anagram? YES
The green and red letters taken together are an anagram for **ODOMETER**.

But you may recall the “wave of emotion” in CEMETERY, where we discussed the waving between the letters M and W.
If the M in ODOMETER becomes a W, then we have an anagram for **WOOD TREE**.
More on this later...

HORSE

There are 4 horses on the map. If these are starting positions we would be justified in looking for 4 letters in the sentences of chapter HORSE that could be used to figuratively propel each of the horses around the map, reading letters in the process. But we would be wrong. The carousel has only 3 riders.

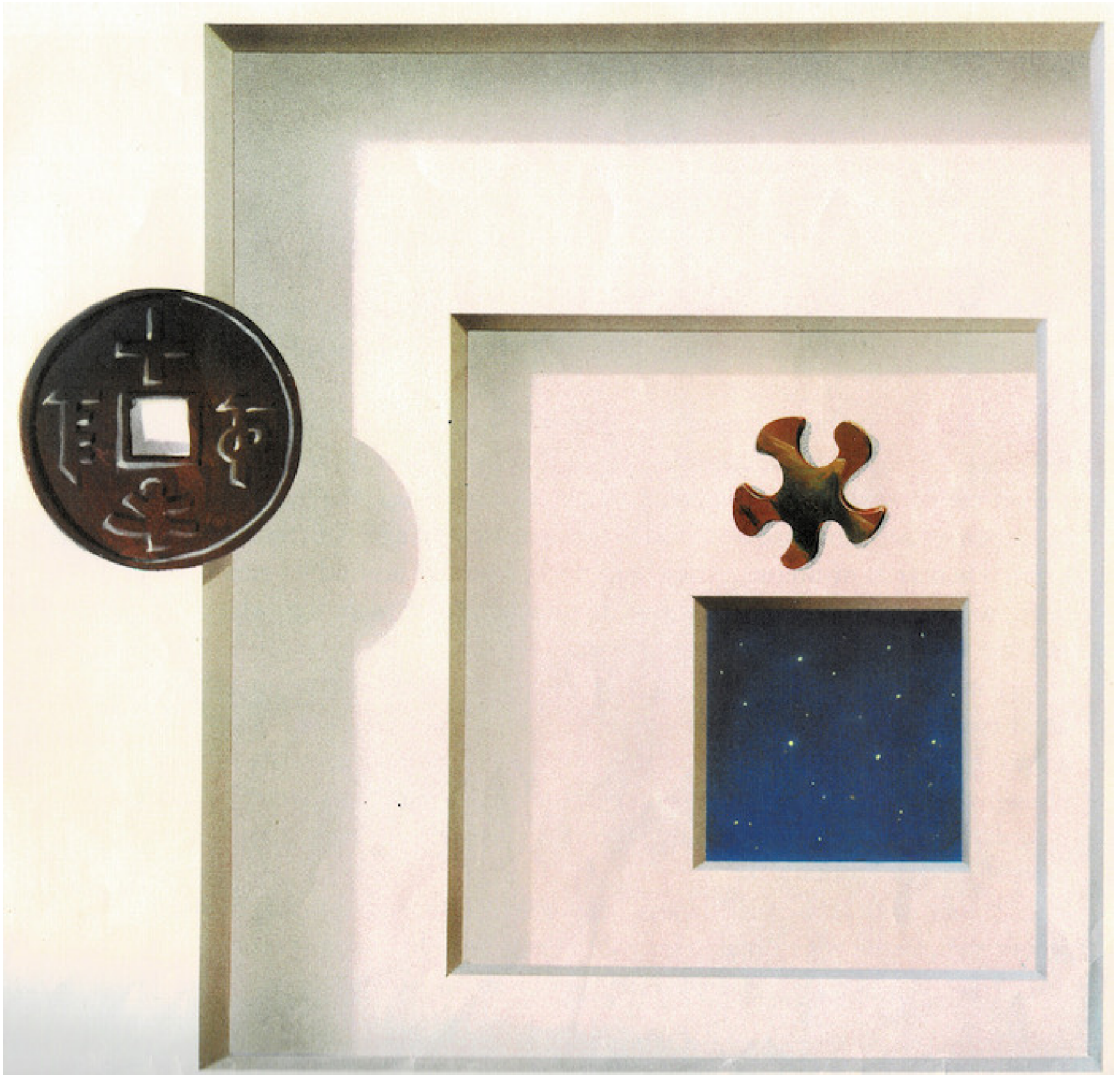
If we look at the original map (the one in 12 pieces), we see that each ARROW has a BAR at one extremity. And the arrows all point to the bottom row of the Journey's text. Even the X could be a sign to **BAR ROW**. This bottom row also contains the words BOOT and BAN. As we will see, the value of letters on this bottom row will **subtract** from the column result, whereas values from the other three sides will **add** to the result.

```
Multicolumn[Characters@"IOPOSBOOTUAIEHONMRSEAEIBAN", {1, 26}, Frame → All,
 Background → {None, None, {{{1, 1}, {6, 9}} → Red, {{1, 1}, {24, 26}} → Red}}]
```

I	O	P	O	S	B	O	O	T	U	A	I	E	H	O	N	M	R	S	E	A	E	I	B	A	N

We are left with 3 horses, and therefore 3 letters necessary to move them independently. In HORSE (chapter 21), the choice is obvious: **OWL** (the "watchman")

On the image above, the missing piece looking like a gear appears in FORTUNE, where the Journey occurs.



Here is what FORTUNE says about the coin, which represent the 4 horses.
“Three coins of ancient age”
“She picked up one”
“It was larger than the others”
“Worn by a million seeking hands”
“She turned it over and found twelve tiny animals around the rim”

The image shows three squares, one inside the other, getting smaller and smaller. These represent **H**ours (or degrees), **M**inutes, **S**econds. Note that when we banned the bottom row of the Journey, we no longer consider the letter E from the word MESH, leaving us with HMS (Hours, Minutes, Seconds). When horses get to move around the map, they will all move at the same time, but at different rate, depending on letters **O W L** in HORSE. But only one of the three letters at a time will be used to match a

letter in the kite string. The choice of the horse (and therefore the letter) will be determined by the column we are working on, as each column will be assigned either to Degrees, Minutes or Seconds.

This image tells us that the horses represented by letters HMS are not associated with Hours, Minutes, Seconds in that order. The seeking hands refer to a watch (this is why we have a watchman to move them). The sentences are about turning and time. The coin is on the larger square (Hour), one that has 12 hours around the clock (like the animals). The gear with 5 spokes indicates a rotation by **5 units clockwise**. We end up with the situation below, on the right side of the arrow, showing which horse is associated with which part of the coordinate after this rotation.

```
Rule[Grid[{Characters@"HMS", {"Hours", "Minutes", "Seconds"}}, Frame → All],
 Grid[{Characters@"HMS", RotateRight[{"Hours", "Minutes", "Seconds"}, 5]},
  Frame → All]]
```

H	M	S
Hours	Minutes	Seconds

→

H	M	S
Minutes	Seconds	Hours

Here we show a representation of the full grid with a red bottom row and three horses.

```
topRowJourney = {"LMULNISITALIMRPEREMIELNLJE"};
sideRowsJourneyRaw = {"N_S", "T_N", "E_T", "R_N", "H_S",
    "I_V", "O_L", "A_E", "N_I", "R_T", "V_E", "N_O", "T_O", "E_L", "T_H",
    "T_M", "K_I", "O_Y", "W_O", "N_E", "O_A", "R_N", "H_E", "N_S"};
bottomRowJourney = {"IOPOSBOOTUAIEHONMRSEAEIBAN"};
cemeteryMap = Partition[Characters@cemeteryWordsP21Mod5, 24];
fullRows  = Join[topRowJourney,
    MapIndexed[StringReplace[#1, "_" → StringJoin@cemeteryMap[[#2]]] &,
     sideRowsJourneyRaw], bottomRowJourney];
Grid[Characters[#] & /@ fullRows, Frame → All, ItemSize → {1, 1},
 Background → {None, 26 → Red, {{6, 1} → Cyan, {6, 26} → Cyan,
     {1, 13} → Cyan, {1, 1} → LightOrange, {1, 26} → LightOrange,
     {26, 1} → Red, {26, 26} → Red, {{2, 25}, {2, 25}} → LightGray,
     {5 + 1, 12 + 1} → White, {22 + 1, 9 + 1} → Lighter@LightGray}}]
```

L	M	U	L	N	I	S	I	T	A	L	I	M	R	P	E	R	E	M	I	E	L	N	L	J	E
N	D	E	A	T	H	I	N	M	A	N	Y	S	I	Z	E	S	S	O	M	E	A	N	A	M	S
T	E	S	J	U	S	T	T	H	R	E	E	T	I	M	E	S	F	O	U	N	D	W	I	T	N
E	H	N	A	M	E	S	W	O	R	N	A	W	A	Y	M	A	N	Y	W	E	R	E	W	I	T
R	T	H	M	E	S	S	A	G	E	S	O	F	L	O	V	E	C	O	B	W	E	B	S	T	N
H	H	E	M	A	M	O	N	G	T	H	E	M	F	L	O	W	E	R	S	R	U	S	T	L	S
I	E	D	I	N	A	W	I	N	D	B	R	O	U	G	H	T	A	S	I	N	G	L	E	R	V
O	O	S	E	T	O	O	N	F	A	T	H	E	R	S	R	E	S	T	I	N	G	P	L	A	L
A	C	E	A	S	F	R	O	M	S	T	O	N	E	L	I	Z	A	R	D	S	T	O	S	T	E
N	O	N	E	R	E	A	D	I	N	G	N	A	M	E	S	M	E	S	S	A	G	E	S	F	I
R	R	I	G	H	T	E	N	E	D	L	O	S	S	O	F	S	T	I	L	L	A	N	U	N	T
V	H	E	A	L	E	D	K	N	U	T	S	E	N	J	U	N	E	M	O	U	N	D	F	E	E
N	L	T	T	H	A	T	E	A	C	H	S	T	E	P	T	O	O	K	T	O	T	O	H	E	O
T	R	S	E	L	F	W	H	E	N	S	M	A	L	L	E	A	C	H	S	T	E	P	M	A	O
E	N	Y	S	T	E	P	S	T	A	K	E	N	I	N	Y	E	A	R	S	G	O	N	E	B	L
T	Y	W	H	E	N	F	O	U	N	D	F	A	T	H	E	R	S	G	R	A	V	E	H	E	H
T	A	R	T	L	E	A	P	T	U	P	I	N	A	S	I	F	C	L	I	M	B	E	D	T	M
K	O	O	F	A	M	O	U	N	T	A	I	N	I	T	A	S	T	O	N	E	L	I	K	E	I
O	A	N	A	R	C	H	B	E	N	T	T	O	R	O	S	E	U	P	O	N	G	R	A	V	Y
W	E	A	T	O	N	C	E	A	L	I	V	E	W	I	T	H	A	N	W	A	V	E	O	F	O
N	E	M	O	T	I	O	N	F	E	E	L	I	N	G	O	F	R	E	L	E	A	S	E	D	E
O	K	N	E	W	T	R	E	A	S	U	R	E	N	E	A	R	L	O	O	K	I	N	G	U	A
R	P	A	S	T	A	L	L	I	O	N	H	E	R	U	N	N	I	N	G	O	N	O	T	H	N
H	E	R	S	I	D	E	O	F	T	R	E	E	S	A	H	E	A	R	D	S	N	O	R	T	E
N	I	N	G	O	F	E	C	H	O	O	F	T	O	T	O	C	O	U	L	D	M	O	V	E	S
I	O	P	O	S	B	O	O	T	U	A	I	E	H	O	N	M	R	S	E	A	E	I	B	A	N

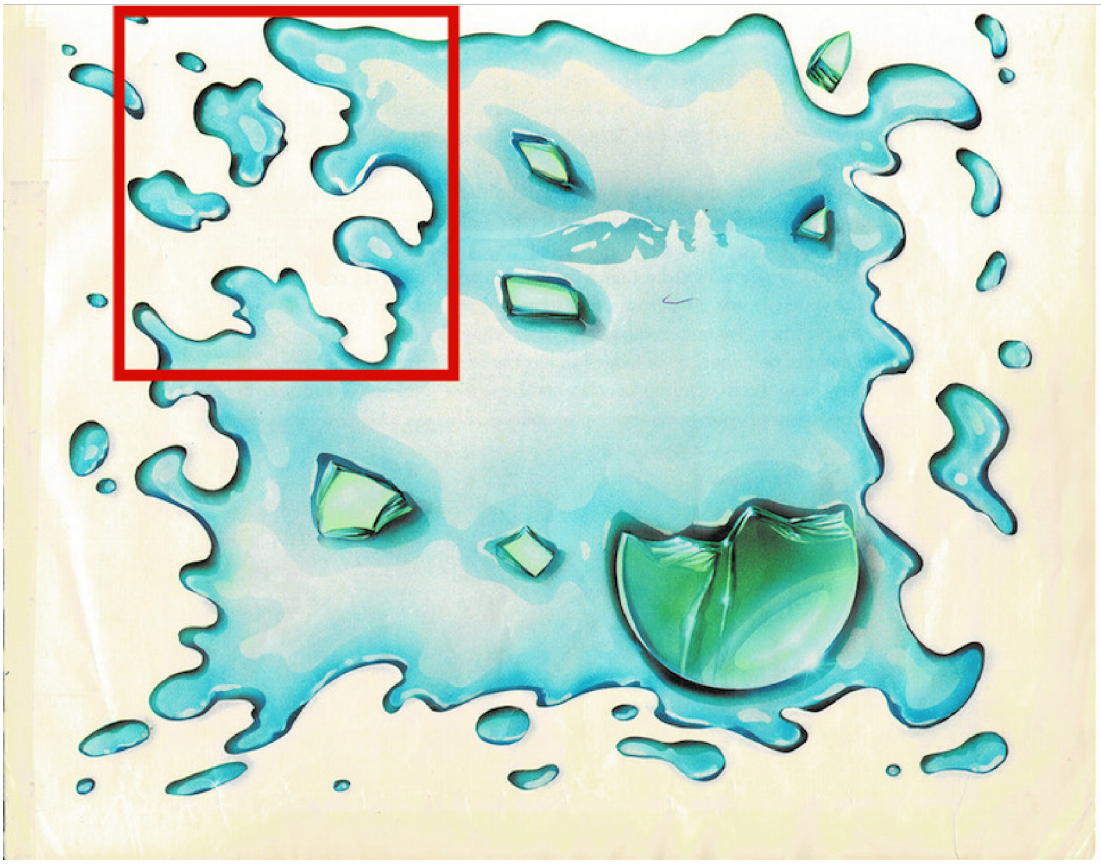
Of course we have to determine which letter in **O W L** drives which of the horses. The answer is in one of the sentences in HORSE, on page 76:

“AN **OWL S**TARTLED **H**ER”

We have the words “ OWL START”. So starting with the word “startled” , letters will appear in the order

matching **O-W-L**. We match **O** to the **S**econds horse, **W** to the **H**ours (degrees) horse, and we are left with **L** matching the **M**inutes horse.

The reason why the small gear shows the image of a mountain is found in the following image in MOUNTAIN.



We find the trace left by the gear found in CREATION, the one we used to derive our “driver”. In fact, sentences in HORSE will be rotated by the same gear as the one rotating the sentences in RIVER. This symbolizes the fact that these movements take place in synchrony. Each new character in our 56-letter string (Kite run #13), generated from a set of directions in RIVER sentences, is accompanied by the three horses moving a set distance determined by HORSE sentences.

We are now ready to go through the text of chapter 21 (HORSE), a process similar as the one used for chapter 15 (RIVER).

```
horseRaw =
  "Amanda followed a stream along the floor of the great forest. The giant
    trees swayed in the wind, and light fell over her like gentle leaves.
    The sound of many waterfalls urged her on. The years had covered the
    rocky hill side with fallen leaves and broken boughs of pine and fir.
    The leaves and boughs had decayed into rich soil, covered now by moss
```

```
and fern that was soft beneath her feet. Each step was a discovery. A
piece of the final puzzle. A doorway in time. A swaying bridge, buit
by hand from wood and rope, took her across a final chiasm. As she
crossed, the air became sweet with the smell of clover and hay. It
was the early summer smell of many years ago. At the forest's edge
she found an old barn. Plants had covered it in an innocent tangle
of vines. But the big double doors still hung firm on their hinges.
And on the doors, she saw handles of iron curled around in a curious
design that quickened her heart. It was his workshop. Her father's old
place. An owl startled her when she opened the doors. He circled the
barn and perched on a nearby tree. He was the watchman who had watched
through the night. She stepped inside. Cool dust lay everywhere, for
here time had truly stopped. Each article lay waiting as if its master
would soon return. She found his anvil and hammer, the cold black
furnace, and an iron melting spoon that was flecked with traces of
gold. On the workbench lay her father's plaid coat. It was coarse to
the touch, but when she picked it up and held it to her face, it felt
warm and full of safety. A chain lay on a clay sculpture of a horse.
When she lifted of the chain and picked up the earthen figure she
suddenly saw, pinned to the wall, drawings that had filled her dream.
They were swirling ink sketches of a spirited horse, his mane flying,
his hoof pawing the air. Through a dusty window, glassed blurred with
time, she saw a magical scene. In the meadow, waiting for her, was
her stallion. He was where he had always been. Amanda ran through the
high grass to Treasure. She knew now that he was hers. He did not bolt
but turned his head to meet her. And when she touched him, his skin
glowed under her stroking hand, and he nuzzled her with his warm,
velvet nose. She felt her father nearby, as if she were a child again.
But now she had grown tall, with strong legs and certain hands. His
fingers twined around Treasure's mane and she was on him in a second.
The wind brought the scent of a faraway sea. She could smell it and
so could he. She talked to him with her hands and he understood. He
turned with a touch and she guided him easily, for she knew the way.
She had dreamt it often. Amanda bent close to him, her hair mingling
with his mane. This was the way they rode all day until the waves
broke at their feet and the sand flew high behind them. The sun slid
down to the ocean. The moon rose, and still they went on together.
For this was just their beginning. And the horse of gold? Amanda did
not need it. She had found her Treasure. Now you find yours. Take all
the clues, hidden herein. Combine them and take their full measure.
It's time for your search to begin. It may lead you to Treasure.";
```

```
horseSentencesRaw = AssociationThread[
   Range[Length[TextSentences[horseRaw]]], TextSentences[horseRaw]];
Column@Normal@horseSentencesRaw
```

```
1 → Amanda followed a stream along the floor of the great forest.
2 →
 The giant trees swayed in the wind, and light fell over her like gentle leaves.
3 → The sound of many waterfalls urged her on.
4 → The years had covered the rocky hill
   side with fallen leaves and broken boughs of pine and fir.
5 → The leaves and boughs had decayed into rich soil,
   covered now by moss and fern that was soft beneath her feet.
6 → Each step was a discovery.
7 → A piece of the final puzzle.
8 → A doorway in time.
9 → A swaying bridge, buit by hand
   from wood and rope, took her across a final chiasm.
10 → As she crossed, the air became sweet with the smell of clover and hay.
11 → It was the early summer smell of many years ago.
12 → At the forest's edge she found an old barn.
13 → Plants had covered it in an innocent tangle of vines.
14 → But the big double doors still hung firm on their hinges.
15 → And on the doors, she saw handles of iron
   curled around in a curious design that quickened her heart.
16 → It was his workshop.
17 → Her father's old place.
18 → An owl startled her when she opened the doors.
19 → He circled the barn and perched on a nearby tree.
20 → He was the watchman who had watched through the night.
21 → She stepped inside.
22 → Cool dust lay everywhere, for here time had truly stopped.
23 → Each article lay waiting as if its master would soon return.
24 → She found his anvil and hammer, the cold black furnace,
   and an iron melting spoon that was flecked with traces of gold.
25 → On the workbench lay her father's plaid coat.
26 → It was coarse to the touch, but when she picked it
   up and held it to her face, it felt warm and full of safety.
27 → A chain lay on a clay sculpture of a horse.
28 → When she lifted of the chain and picked up the earthen figure she
   suddenly saw, pinned to the wall, drawings that had filled her dream.
29 → They were swirling ink sketches of a
   spirited horse, his mane flying, his hoof pawing the air.
30 → Through a dusty window, glassed blurred with time, she saw a magical scene.
31 → In the meadow, waiting for her, was her stallion.
32 → He was where he had always been.
33 → Amanda ran through the high grass to Treasure.
```

```
34 → She knew now that he was hers.
35 → He did not bolt but turned his head to meet her.
36 → And when she touched him, his skin glowed under her
    stroking hand, and he nuzzled her with his warm, velvet nose.
37 → She felt her father nearby, as if she were a child again.
38 → But now she had grown tall, with strong legs and certain hands.
39 → His fingers twined around Treasure's mane and she was on him in a second.
40 → The wind brought the scent of a faraway sea.
41 → She could smell it and so could he.
42 → She talked to him with her hands and he understood.
43 → He turned with a touch and she guided him easily, for she knew the way.
44 → She had dreamt it often.
45 → Amanda bent close to him, her hair mingling with his mane.
46 → This was the way they rode all day until the
    waves broke at their feet and the sand flew high behind them.
47 → The sun slid down to the ocean.
48 → The moon rose, and still they went on together.
49 → For this was just their beginning.
50 → And the horse of gold?
51 → Amanda did not need it.
52 → She had found her Treasure.
53 → Now you find yours.
54 → Take all the clues, hidden herein.
55 → Combine them and take their full measure.
56 → It's time for your search to begin.
57 → It may lead you to Treasure.
```

We go through these 57 sentences to generate sets of distances traveled by horses, using the counts of letters **O-W-L** in each sentence.
In the resulting sets, we convert the letters **O-W-L** to **S-H-M**, to match distances with the proper horse (representing Hours, Minutes, Seconds).

```
owlDirectionsRaw = <| |>;
(
    sent = horseSentencesRaw[[Key[#]]];
    AppendTo[owlDirectionsRaw,
     # → LetterCounts@StringReplace[StringDelete[ToUpperCase@sent,
         Except[Characters@"OWL"]], {"O" → "S", "W" → "H", "L" → "M"}]];

   ) & /@ Range[57];
Column@Normal@owlDirectionsRaw
```

```
1 → <| S → 7, M → 4, H → 1 |>
2 → <| M → 6, H → 2, S → 1 |>
3 → <| S → 3, M → 2, H → 1 |>
4 → <| M → 5, S → 5, H → 1 |>
5 → <| S → 7, H → 2, M → 2 |>
6 → <| S → 1, H → 1 |>
```

```
7 → ⟨|M → 2, S → 1|⟩
8 → ⟨|S → 2, H → 1|⟩
9 → ⟨|S → 7, H → 2, M → 1|⟩
10 → ⟨|M → 3, S → 3, H → 2|⟩
11 → ⟨|M → 3, S → 2, H → 1|⟩
12 → ⟨|S → 3, M → 1|⟩
13 → ⟨|S → 3, M → 2|⟩
14 → ⟨|S → 4, M → 3|⟩
15 → ⟨|S → 7, M → 2, H → 1|⟩
16 → ⟨|S → 2, H → 2|⟩
17 → ⟨|M → 2, S → 1|⟩
18 → ⟨|S → 4, M → 2, H → 2|⟩
19 → ⟨|S → 1, M → 1|⟩
20 → ⟨|H → 4, S → 2|⟩
21 → ⟨| |⟩
22 → ⟨|S → 4, M → 3, H → 1|⟩
23 → ⟨|S → 3, M → 3, H → 2|⟩
24 → ⟨|S → 7, M → 6, H → 2|⟩
25 → ⟨|S → 3, M → 2, H → 1|⟩
26 → ⟨|S → 5, M → 4, H → 3|⟩
27 → ⟨|S → 3, M → 3|⟩
28 → ⟨|M → 6, H → 4, S → 2|⟩
29 → ⟨|S → 4, H → 3, M → 2|⟩
30 → ⟨|H → 4, M → 3, S → 2|⟩
31 → ⟨|H → 3, S → 3, M → 2|⟩
32 → ⟨|H → 3, M → 1|⟩
33 → ⟨|S → 2|⟩
34 → ⟨|H → 3, S → 1|⟩
35 → ⟨|S → 3, M → 1|⟩
36 → ⟨|S → 4, H → 4, M → 3|⟩
37 → ⟨|M → 2, H → 1|⟩
38 → ⟨|M → 3, H → 3, S → 3|⟩
39 → ⟨|S → 3, H → 2|⟩
40 → ⟨|S → 2, H → 2|⟩
41 → ⟨|M → 4, S → 3|⟩
42 → ⟨|S → 3, H → 1, M → 1|⟩
43 → ⟨|H → 3, S → 2, M → 1|⟩
44 → ⟨|S → 1|⟩
45 → ⟨|S → 2, M → 2, H → 1|⟩
46 → ⟨|M → 4, H → 4, S → 2|⟩
47 → ⟨|S → 3, H → 1, M → 1|⟩
48 → ⟨|S → 5, M → 2, H → 1|⟩
49 → ⟨|H → 1, S → 1|⟩
50 → ⟨|S → 3, M → 1|⟩
51 → ⟨|S → 1|⟩
52 → ⟨|S → 1|⟩
53 → ⟨|S → 3, H → 1|⟩
54 → ⟨|M → 3|⟩
55 → ⟨|M → 2, S → 1|⟩
56 → ⟨|S → 3|⟩
57 → ⟨|S → 2, M → 1|⟩
```

We create the function *getFinalMoves* to apply the CREATION driver twice to the above sentences to produce the final list of 56 sets of horse moves, properly ordered.

```
getFinalMoves[driver_, rawDirections_] := Module[{current, visited, func},
  current = 0;
  visited = {};
  (
     If[# == "V", func = Plus];
     If[# == "F", func = Subtract];
     sum = func[current, LetterNumber[#]];
     If[sum > 56, sum = sum - 56];
     If[sum < 1, sum = 56 + sum];
     AppendTo[visited, sum];
     current = sum;
    ) & /@ driver;
  rawDirections[[#]] & /@ visited
 ]
```

Sentence #21 does not contain any **O W L** directions . “She stepped inside” signifies **sidestepped**. This sentence is removed before calling the function, bringing the number of sets to 56.

```
owlFinalDirections = getFinalMoves[
  Join[creationDriver, creationDriver], Delete[owlDirectionsRaw, 21]]
```

```
{⟨|S → 3, M → 3, H → 2|⟩, ⟨|S → 2, H → 2|⟩, ⟨|M → 3, S → 2, H → 1|⟩, ⟨|S → 2, H → 1|⟩,
 ⟨|S → 3, H → 1, M → 1|⟩, ⟨|M → 4, H → 4, S → 2|⟩, ⟨|M → 6, H → 4, S → 2|⟩, ⟨|S → 2, H → 1|⟩,
 ⟨|S → 7, H → 2, M → 2|⟩, ⟨|S → 5, M → 2, H → 1|⟩, ⟨|S → 3, H → 1, M → 1|⟩, ⟨|S → 2, H → 2|⟩,
 ⟨|S → 3, H → 2|⟩, ⟨|S → 3, M → 1|⟩, ⟨|M → 2, S → 1|⟩, ⟨|S → 3, M → 1|⟩, ⟨|S → 3, M → 2|⟩,
 ⟨|M → 3, S → 3, H → 2|⟩, ⟨|S → 1|⟩, ⟨|S → 2, M → 2, H → 1|⟩, ⟨|S → 2, H → 2|⟩,
 ⟨|S → 3, M → 3|⟩, ⟨|S → 3, M → 1|⟩, ⟨|S → 7, H → 2, M → 1|⟩, ⟨|S → 2, H → 1|⟩, ⟨|M → 3|⟩,
 ⟨|S → 1|⟩, ⟨|H → 4, M → 3, S → 2|⟩, ⟨|S → 1|⟩, ⟨|M → 4, H → 4, S → 2|⟩, ⟨|M → 4, S → 3|⟩,
 ⟨|M → 3, H → 3, S → 3|⟩, ⟨|S → 1, M → 1|⟩, ⟨|S → 4, M → 2, H → 2|⟩, ⟨|S → 2, M → 1|⟩,
 ⟨|M → 3, H → 3, S → 3|⟩, ⟨|S → 3, M → 1|⟩, ⟨|H → 4, S → 2|⟩, ⟨|S → 1, M → 1|⟩,
 ⟨|S → 3, M → 1|⟩, ⟨|M → 3, S → 2, H → 1|⟩, ⟨|S → 3, H → 1, M → 1|⟩, ⟨|M → 6, H → 4, S → 2|⟩,
 ⟨|M → 2, S → 1|⟩, ⟨|H → 3, S → 2, M → 1|⟩, ⟨|S → 2, H → 2|⟩, ⟨|S → 3, M → 2, H → 1|⟩,
 ⟨|M → 2, S → 1|⟩, ⟨|S → 3, M → 1|⟩, ⟨|S → 3|⟩, ⟨|S → 3, H → 1, M → 1|⟩, ⟨|S → 3, H → 2|⟩,
 ⟨|M → 3, H → 3, S → 3|⟩, ⟨|S → 3, M → 3|⟩, ⟨|S → 7, M → 6, H → 2|⟩, ⟨|M → 6, H → 2, S → 1|⟩}
```

We have said that we will know which one of the three horses (H,M, or S) contributes a letter to match the current letter generated in the kite’s string of 56 letters by looking at the column where this last letter appears. This is possible because each of our 24 grid columns will be assigned one of three possible role: a **degrees column**, a **minutes column** or a **seconds column**. How is this assignment done? When it comes to degrees, minutes and seconds used to represent latitude and longitude, there is only one person who can help us: **MR. MAPS**.

MR. MAPS

What is immediately striking about this chapter is the sheer number of curly single and double quotes. In expressing latitude and longitude a single quote is a symbol for minutes and the closing double quote is a symbol for seconds. What about degrees? A degree symbol is like the letter **o** placed high on the line of text. It has been cleverly inserted in MR MAPS under the form of the digraph **IO** (high o) found in a few words.

We start with the raw text of MR. MAPS (chapter 9).

```
mrMapsRaw =
  "The town at the edge of the desert, like an oasis, had grown around springs
    of water that bubbled from the heart of the earth. Here groves of
    plants and trees flourished. In this town was a keeper of maps. The
    innkeeper’s wife told Amanda about him. He lived in a tiny house behind
    a grove of bamboo and far from the main road. By the time she found
    it, night had once again fallen. His house was very odd, with the
    blinds all pulled down, and who was to say what he was like? amanda
    was hesitant to ask the man for help, but she had come this far. She
    could go a little farther. In fact, the door cracked open before she
    could even knock. A wizened old man, who did not come up to her chin,
    stared at her with extraordinary blue eyes. He knew immediately what
    she wanted. “Lost?” He did not wait for an answer. The door opened
    wide and she saw an amazing collection of maps. Black and white maps,
    color maps, maps of states and streets and counties and cities. Maps
    of mountains and maps of valleys. Maps of every American place, and
    even of the sky. Maps, mostly old, with curly edges and finely drawn
    lines. Maps filled the little house to overflowing. There were maps in
    boxes, maps in baskets, maps on the walls, and maps on the ceiling.
    (And even, she thought for a moment, maps floating loose in the air.)
    “Lots of people lost nowadays,” said Mr. Maps, stepping back to let her
    in. “Can’t find the places they’re looking for.” Amanda said nothing,
    but it didn’t matter. Mr. Maps was used to talking to himself. “You
    gotta have a map.” He admired his collection with obvious satisfaction.
    “Any place you want to go, gotta have one. ’Course the thing is” he
    turned to her whispering in a low voice reserved for secrets, “you
    got to know where you’re going.” “Lots of places. Oh, my, lots of
    places you could go. But I got the maps for all of them.” He opened
    a trunk to show her even more. But lying in the trunk on a pile of
    maps was a black falcon. Amanda couldn’t tell whether it was alive
    or just a statue. “Yes, ma’am, can’t have too many maps,” said Mr.
    Maps. He hopped around trying to distract her as he hastily covered
    the dingus with even more maps and slammed shut the trunk. “’Cause you
```

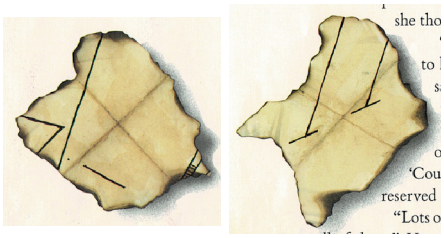
```
never know what place you're gonna have to find." "I got all these
degrees and meridians...south of east...north of west...any direction
you want to go." Any direction she wanted to go? She'd come all this
way hoping the funny old man would tell her the direction she should
take, and now he expected her to know where she was going. She felt
so confused that the room started to spin. She and Mr. Maps turned
into a globe, revolving like a planet lost forever among the stars.
"Take these maps," he said. "Trust me. You'll find what you're lookin'
for." Amanda realized it was time to leave. She took the maps from his
hands. And as she stepped outside, she turned to stare one last time
into his enormous blue eyes. Before Mr. Maps could say "Good luck,"
three maps darted through his doorway and escaped into the night.";
```

```
(* Raw sentences in MR. MAPS *)

mrMapsSentencesRaw = TextSentences[mrMapsRaw];
Column@MapIndexed[Row[{#2[[1]], "- ", #1}] &, mrMapsSentencesRaw]
```

```
1- The town at the edge of the desert, like an oasis, had grown
   around springs of water that bubbled from the heart of the earth.
2- Here groves of plants and trees flourished.
3- In this town was a keeper of maps.
4- The innkeeper's wife told Amanda about him.
5- He lived in a tiny house behind a grove of bamboo and far from the main road.
6- By the time she found it, night had once again fallen.
7- His house was very odd, with the blinds
   all pulled down, and who was to say what he was like?
8- amanda was hesitant to ask the man for help, but she had come this far.
9- She could go a little farther.
10- In fact, the door cracked open before she could even knock.
11- A wizened old man, who did not come up
   to her chin, stared at her with extraordinary blue eyes.
12- He knew immediately what she wanted.
13- "Lost?"
14- He did not wait for an answer.
15- The door opened wide and she saw an amazing collection of maps.
16- Black and white maps, color maps,
   maps of states and streets and counties and cities.
17- Maps of mountains and maps of valleys.
18- Maps of every American place, and even of the sky.
19- Maps, mostly old, with curly edges and finely drawn lines.
20- Maps filled the little house to overflowing.
21- There were maps in boxes, maps
   in baskets, maps on the walls, and maps on the ceiling.
22- (And even, she thought for a moment, maps floating loose in the air.)
```

```
23- “Lots of people lost nowadays,” said Mr. Maps, stepping back to let her in.
24- “Can’t find the places they’re looking for.”
25- Amanda said nothing, but it didn’t matter.
26- Mr. Maps was used to talking to himself.
27- “You gotta have a map.”
28- He admired his collection with obvious satisfaction.
29- “Any place you want to go, gotta have one.
30- ’Course the thing is” he turned to her whispering in a low
   voice reserved for secrets, “you got to know where you’re going.”
31- “Lots of places.
32- Oh, my, lots of places you could go.
33- But I got the maps for all of them.”
34- He opened a trunk to show her even more.
35- But lying in the trunk on a pile of maps was a black falcon.
36- Amanda couldn’t tell whether it was alive or just a statue.
37- “Yes, ma’am, can’t have too many maps,” said Mr. Maps.
38- He hopped around trying to distract her as he hastily
   covered the dingus with even more maps and slammed shut the trunk.
39- “’Cause you never know what place you’re gonna have to find.”
40- “I got all these degrees and meridians…south
   of east…north of west…any direction you want to go.”
41- Any direction she wanted to go?
42-
 She’d come all this way hoping the funny old man would tell her the direction
   she should take, and now he expected her to know where she was going.
43- She felt so confused that the room started to spin.
44- She and Mr. Maps turned into a globe,
   revolving like a planet lost forever among the stars.
45- “Take these maps,” he said.
46- “Trust me.
47- You’ll find what you’re lookin’ for.”
48- Amanda realized it was time to leave.
49- She took the maps from his hands.
50- And as she stepped outside, she
   turned to stare one last time into his enormous blue eyes.
51- Before Mr. Maps could say “Good luck,” three
   maps darted through his doorway and escaped into the night.
```

In this chapter we have a sentence selector. It is given to us by the map images appearing in the chapter.



The focus of these two images is on the folds in the map, looking like the letter X. Indeed, we need to select sentences containing the letters **F-O-L-D**. The following does the selection and then converts the digraph **IO** to a degree symbol. We are left with these numbered sentences.

```
mrMapsSentencesFOLD =
  MapIndexed[If[StringContainsQ[#1, "F", IgnoreCase → True] && StringContainsQ[#1,
       "O", IgnoreCase → True] && StringContainsQ[#1, "L", IgnoreCase → True] &&
      StringContainsQ[#1, "D", IgnoreCase → True],
     Rule[#2[[1]], StringReplace[#1, "io" → "°"]], Nothing] &, mrMapsSentencesRaw];
Column@mrMapsSentencesFOLD
```

```
1 → The town at the edge of the desert, like an oasis, had grown
   around springs of water that bubbled from the heart of the earth.
2 → Here groves of plants and trees flourished.
4 → The innkeeper’s wife told Amanda about him.
5 → He lived in a tiny house behind a grove of bamboo and far from the main road.
6 → By the time she found it, night had once again fallen.
8 → amanda was hesitant to ask the man for help, but she had come this far.
9 → She could go a little farther.
10 → In fact, the door cracked open before she could even knock.
15 → The door opened wide and she saw an amazing collect°n of maps.
16 → Black and white maps, color maps,
   maps of states and streets and counties and cities.
17 → Maps of mountains and maps of valleys.
18 → Maps of every American place, and even of the sky.
19 → Maps, mostly old, with curly edges and finely drawn lines.
20 → Maps filled the little house to overflowing.
22 → (And even, she thought for a moment, maps floating loose in the air.)
23 → “Lots of people lost nowadays,” said Mr. Maps, stepping back to let her in.
24 → “Can’t find the places they’re looking for.”
26 → Mr. Maps was used to talking to himself.
28 → He admired his collect°n with obv°us satisfact°n.
30 → ’Course the thing is” he turned to her whispering in a low
   voice reserved for secrets, “you got to know where you’re going.”
32 → Oh, my, lots of places you could go.
39 → “’Cause you never know what place you’re gonna have to find.”
40 → “I got all these degrees and meridians…south
   of east…north of west…any direct°n you want to go.”
42 →
 She’d come all this way hoping the funny old man would tell her the direct°n
   she should take, and now he expected her to know where she was going.
43 → She felt so confused that the room started to spin.
44 → She and Mr. Maps turned into a globe,
   revolving like a planet lost forever among the stars.
47 → You’ll find what you’re lookin’ for.”
51 → Before Mr. Maps could say “Good luck,” three
   maps darted through his doorway and escaped into the night.
```

Sentence #51 tells us that the three maps (degrees, minutes, seconds) are completed before Mr. Maps says “Good luck”. So the last curly double quotes in sentence #51 is not to be counted. Sentence #47

tells us that we can find what we want at this point.
Note that sentence #28 has three degree symbols. This is why it mentions a “collection”.
Sentences #34 to #38, with the trunk (truncate) and the black bird, are not included in the list of selected sentences.

We now collect the degree, minute and second symbols, skipping sentence #51. We must keep track of the sentences corresponding to these symbols because of possible specific rules in the ways to deal with them.

```
snList = {};
symbolsLat = {};

(
    sNumber = #[[1]];
    sText = Characters@StringDelete[#[[2]], Except[{"’", "”", "°"}]];
    i = 1;
    While[i ≤ Length[sText],
     If[MemberQ[snList, sNumber],
      AppendTo[snList, SpanFromLeft], AppendTo[snList, sNumber]];
     AppendTo[symbolsLat, sText[[i]]];
     i++;
    ]

   ) & /@ mrMapsSentencesFOLD[[1 ;; 27]];
symbolsLon = RotateRight[symbolsLat];
symbolMap =
 Grid[{Join[{"Sentence number"}, snList], Join[{"Latitude"}, symbolsLat],
   Join[{"Longitude"}, symbolsLon], Join[{"Map column"}, Range[24]]}, Frame → All,
  Background → {1 → LightGray, None, {{4, 5} → Green, {4, 14} → Green, {4, 19} → Green,
     {4, 21} → Green, {4, 2} → Red, {4, 7} → Red, {4, 9} → Red, {4, 17} → Red}}]
```

Sentence number	4	15	23	24			28			30				39			40		42		47			
Latitude	’	◦	”	’	’	”	◦	◦	◦	’	”	’	”	’	’	”	◦	”	’	◦	’	’	’	”
Longitude	”	’	◦	”	’	’	”	◦	◦	◦	’	”	’	”	’	’	”	◦	”	’	◦	’	’	’
Map column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

This table is very busy. We have added the map columns corresponding to the green kite (latitude) and red kite (longitude) identified earlier.
The latitude row shows the straightforward order of symbols appearing in the sentences numbered in the first row.

Why a different longitude row? This row is the same as the latitude row rotated right by one. This rotation is alluded to in sentence #1: “springs of water that bubbled from the heart of the earth”. From the EARTH to the HEART is a rotation of letters by one. But this rotation occurs starting in sentence #43,

because this is where sentence #1 takes its full sense: The following appears in sentence #43: "**EA**ST-...NO**RTH**". Here we have the EARTH in bold with a desert in the middle. And sentence #43 is where we find "degrees and meridians" (**not parallels**). This would indicate that the rotation of symbols applies to longitude. But sentence #43, in concert with sentence #1, also contains an extraordinary clue that we will cover in the next section.

Note that the green kite uses 2 columns to compute latitude seconds (13-18), whereas the red kite uses 2 columns to compute longitude minutes (6-16). This is why we have in both cases 4 columns to measure the three components of the coordinates.

There are notable specific rules applicable to the result of columns found in specific sentences:

- Sentence #4 deals with the "innkeeper's wife". Bringing up the wife is a way to refer to the "other half" or "better half". We will need to half the value found in column #1
- Sentence #24 refers to the inability to find the **places** we are looking for. The places here refer to the position of digits in a number. It looks like in columns 4 and 6, both dealing with minutes, the position of the digits in the result obtained will have to be reversed.
- Sentence #28 deals with a **collection** of degrees. In column #8, the word collection is actually selected. So the degrees of longitude gets to be computed with the collection of degrees in this sentence. It therefore includes column #8 and #9.
- Sentence #39 again deals with places but it is more clever than sentence #24. We will cover it when we get to columns 16 in the next section.

COMPUTING LATITUDE AND LONGITUDE

We now have all the required information to compute latitude and longitude. We could make a long *Mathematica* program that includes all this information and spits out result, but we will instead break down the process to make it easier to follow what is going on.

The following function takes the list of moves directed by the owl and the journey's text. It returns which letter each of the horses sits on at each of the 56 steps.

```
horseRide[moves_, journey_] := Module[{},
  sInitial = 66;
  hInitial = 31;
  mInitial = 48;
  sPosRaw = sInitial + Accumulate@
     Flatten@(Values@KeyTake[#, {"S"}] & /@ owlFinalDirections /. {} → {0});
  mPosRaw = mInitial + Accumulate@
     Flatten@(Values@KeyTake[#, {"M"}] & /@ owlFinalDirections /. {} → {0});
  hPosRaw = hInitial + Accumulate@
     Flatten@(Values@KeyTake[#, {"H"}] & /@ owlFinalDirections /. {} → {0});
  hPos = ReplaceAll[QuotientRemainder[#, 100]⟦2⟧ & /@ hPosRaw, 0 → 100];
  mPos = ReplaceAll[QuotientRemainder[#, 100]⟦2⟧ & /@ mPosRaw, 0 → 100];
  sPos = ReplaceAll[QuotientRemainder[#, 100]⟦2⟧ & /@ sPosRaw, 0 → 100];
  hLetters = If[1 ≤ # ≤ 11, Style[journey⟦#⟧, Red], journey⟦#⟧] & /@ hPos;
  mLetters = If[1 ≤ # ≤ 11, Style[journey⟦#⟧, Red], journey⟦#⟧] & /@ mPos;
  sLetters = If[1 ≤ # ≤ 11, Style[journey⟦#⟧, Red], journey⟦#⟧] & /@ sPos;
  Grid[{Join[{" "}, Range[56]], Join[{"Minutes"}, hLetters],
    Join[{"Seconds"}, mLetters], Join[{"Hours"}, sLetters]},
   Frame → All, Background → {1 → LightGray, 1 → LightGray}]
 ]
```

We call the function and get a grid showing the positions of horses. Red letters indicate the bottom side of the map (negative values). Note that the result displayed here is made up of two png images derived from the actual Mathematica result, as the original grid was too wide to display in the pdf document.

```
horseRide[owlFinalDirections, journeyStringLetters]
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Minutes	E	N	L	M	U	S	L	I	R	P	E	E	I	I	I	I	I	L	L	N	J	J	J	S	N	N	N	V
Seconds	E	E	M	M	I	L	N	N	V	E	I	I	I	T	O	O	H	Y	Y	E	E	E	S	N	N	I	I	E
Hours	E	T	O	L	I	O	A	E	A	N	E	A	O	S	O	I	R	W	O	T	E	V	A	N	M	M	U	N

29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
V	T	T	O	O	H	H	Y	Y	N	N	N	E	S	I	I	E	R	M	M	M	M	N	H	A	A	T	O
E	N	I	T	O	B	S	O	I	I	N	H	N	W	T	V	R	R	A	I	H	H	R	R	N	U	T	P
I	I	L	R	P	M	E	L	S	T	N	L	I	O	L	H	I	O	N	E	A	E	S	N	E	U	P	O

We have already determined what the string in kite #13 looks like.

k1k13

D	E	A	T	H	I	N	M	A	N	Y	S	I	Z	E	S	S	O	M	E	A	N	A	M
E	S	J	U	S	T	T	H	R	E	E	T	I	M	E	S	F	O	U	N	D	W	I	T
H	N	A	M	E	S	W	O	R	N	A	W	A	Y	M	A	N	Y	W	E	R	E	W	I
T	H	M	E	S	S	A	G	E	S	O	F	L	O	V	E	C	O	B	W	E	B	S	T
H	E	M	A	M	O	N	G	T	H	E	M	F	L	O	W	E	R	S	R	U	S	T	L
E	D	I	N	A	W	I	N	D	B	R	O	U	G	H	T	A	S	I	N	G	L	E	R
O	S	E	T	O	O	N	F	A	T	H	E	R	S	R	E	S	T	I	N	G	P	L	A
C	E	A	S	F	R	O	M	S	T	O	N	E	L	I	Z	A	R	D	S	T	O	S	T
O	N	E	R	E	A	D	I	N	G	N	A	M	E	S	M	E	S	S	A	G	E	S	F
R	I	G	H	T	E	N	E	D	L	O	S	S	O	F	S	T	I	L	L	A	N	U	N
H	E	A	L	E	D	K	N	U	T	S	E	N	J	U	N	E	M	O	U	N	D	F	E
L	T	T	H	A	T	E	A	C	H	S	T	E	P	T	O	O	K	T	O	T	O	H	E
R	S	E	L	F	W	H	E	N	S	M	A	L	L	E	A	C	H	S	T	E	P	M	A
N	Y	S	T	E	P	S	T	A	K	E	N	I	N	Y	E	A	R	S	G	O	N	E	B
Y	W	H	E	N	F	O	U	N	D	F	A	T	H	E	R	S	G	R	A	V	E	H	E
A	R	T	L	E	A	P	T	U	P	I	N	A	S	I	F	C	L	I	M	B	E	D	T
O	O	F	A	M	O	U	N	T	A	I	N	I	T	A	S	T	O	N	E	L	I	K	E
A	N	A	R	C	H	B	E	N	T	T	O	R	O	S	E	U	P	O	N	G	R	A	V
E	A	T	O	N	C	E	A	L	I	V	E	W	I	T	H	A	N	W	A	V	E	O	F
E	M	O	T	I	O	N	F	E	E	L	I	N	G	O	F	R	E	L	E	A	S	E	D
K	N	E	W	T	R	E	A	S	U	R	E	N	E	A	R	L	O	O	K	I	N	G	U
P	A	S	T	A	L	L	I	O	N	H	E	R	U	N	N	I	N	G	O	N	O	T	H
E	R	S	I	D	E	O	F	T	R	E	E	S	A	H	E	A	R	D	S	N	O	R	T
I	N	G	O	F	E	C	H	O	O	F	T	O	T	O	C	O	U	L	D	M	O	V	E

And finally we have our symbol maps.

symbolMap

Sentence number	4	15	23	24			28			30				39			40		42		47			
Latitude	’	◦	”	’	’	”	◦	◦	◦	’	”	’	”	’	’	”	◦	”	’	◦	’	’	’	”
Longitude	”	’	◦	”	’	’	”	◦	◦	◦	’	”	’	”	’	’	”	◦	”	’	◦	’	’	’
Map column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Given a column number, the following function will extract the coordinates matching a particular column, together with its position in the list of 56 coordinates. The position can be used to find out what horse letter applies.

```
getColumnCoordinates[n_] :=
 Select[AssociationThread[Range[56], visitedCoordinates[[13]]], #[[2]] == n &]
```

Here is an example of using the function for column 6.

```
getColumnCoordinates[6]
```

```
<| 14 → {13, 6}, 19 → {11, 6}, 24 → {11, 6} |>
```

Using this example, here are the steps to follow:

- Look for column 6 on the symbol grid. It applies to *minutes* for longitude (red kite)
- Looking at the right side of the arrows, get the letters corresponding to the three coordinates {13, 6}, {11, 6}, {11, 6} on the cemetery grid. We get in order W, D, D (red squares are counted twice)
- Looking at the left side of the arrows, look at the *minutes* row in columns 14, 19 and 24 of the horse position grid. We get letters I, L, S
- Add all the letter values together I+L+S+W+D+D = 71. If a letter comes out red on the horse grid, it is on the bottom side of the journey's letters and should be subtracted instead of added.
- On the symbol map, read the applicable sentence in MR. MAPS (using the sentence number), corresponding to the selected column to determine if a specific rule must be applied. There is such a rule in this case and we will discuss it below when we compute the actual longitude.

Latitude (the green kite)

NOTE: recall that the actual green squares indicate columns but are not part of the coordinates to be counted in computing latitude.

Seconds

Column #13 and #18 correspond to seconds of latitude.
Get the coordinates visited in these two columns:

```
getColumnCoordinates[13]
```

```
<| 6 → {11, 13}, 39 → {1, 13} |>
```

```
getColumnCoordinates[18]
```

```
<| 5 → {11, 18}, 17 → {24, 18} |>
```

The letter on the grid for {11,13} is **N**. The seconds horse on move 6 is **L**.
The letter on the grid for {1,13} is **I**. The seconds horse on move 39 is **N**.

The letter on the grid for {11,18} is **M**. The seconds horse on move 5 is **I**.
The letter on the grid for {24,18} is **U**. The seconds horse on move 17 is **H**.

We can sum the letters N S I N M E U S to get a value for these columns.

```
secondsLatitudeRaw = Total@(LetterNumber[#] & /@ Characters@"NLINMIUH")
```

```
100
```

60 seconds are used to make an extra minute. We are left with **40 seconds**.

Minutes

```
getColumnCoordinates[4]
```

```
⟨|8 → {12, 4}, 33 → {20, 4}, 41 → {6, 4}|⟩
```

The letter on the grid for {12, 4} is **H** . The minutes horse on move 8 is **I** .
The letter on the grid for {20,4} is **T** . The minutes horse on move 33 is **O** .
The letter on the grid for {6,4} is **N** . The minutes horse on move 41 is **E** .

```
minutesLatitudeRaw = Total@(LetterNumber[#] & /@ Characters@"HITONE")
```

```
71
```

We add the extra minute obtained for the seconds computation. So we have **72** minutes. We remove 60 minutes to make an extra degree and we are left with **12** minutes. But we see on our symbol's map that column #4 falls under the special rule for sentence #24 and the "places". The digits must be inverted.
So the final result is **21 minutes**.

Degrees

```
getColumnCoordinates[20]
```

```
⟨|21 → {17, 20}, 25 → {12, 20}|⟩
```

The letter on the grid for {17,20} is **E** . The degrees horse on move 21 is **E** .
The letter on the grid for {12,20} is **O** . The degrees horse on move 25 is **M** .

```
degreesLatitudeRaw = Total@(LetterNumber[#] & /@ Characters@"EEOM")
```

```
38
```

We add the extra degree collected with the minutes and get **39 degrees**.

LATITUDE: 39°21'40"

Longitude (the red kite)

NOTE: recall that the actual red squares indicate columns, and are counted twice in computing longitude since they are visited twice in the 56-characters string.

Seconds

```
getColumnCoordinates[1]
```

```
⟨|20 → {17, 1}, 30 → {20, 1}, 34 → {20, 1}, 55 → {18, 1}|⟩
```

The letter on the grid for {17,1} is **O** . The seconds horse on move 20 is **E** .
The letter on the grid for {20,1} is **E** . The seconds horse on move 30 is **N** .
The letter on the grid for {20,1} is **E** . The seconds horse on move 34 is **B** .
The letter on the grid for {18,1} is **A** . The seconds horse on move 55 is **T** .

```
secondsLongitudeRaw = Total@(LetterNumber[#] & /@ Characters@"OEEAENT") -
  Total@(LetterNumber[#] & /@ Characters@"B")
```

```
63
```

Here we must apply the specific rule for sentence #4. We will take the “better half”, which is **32 seconds**.

Minutes

Minutes are determined by columns #16 and #6

```
getColumnCoordinates[16]
```

```
⟨|11 → {12, 16}, 12 → {13, 16}, 13 → {12, 16}, 28 → {23, 16}|⟩
```

```
getColumnCoordinates[6]
```

```
<|14 → {13, 6}, 19 → {11, 6}, 24 → {11, 6}|>
```

COLUMN 16

The letter on the grid for {12, 16} is **O** . The minutes horse on move 11 is **E** .
The letter on the grid for {13,16} is **A** . The minutes horse on move 12 is **E** .
The letter on the grid for {12,16} is **O** . The minutes horse on move 13 is **I** .
The letter on the grid for {23,16} is **E** . The minutes horse on move 28 is **V** .

COLUMN 6

The letter on the grid for {13,6} is **W** . The minutes horse on move 14 is **I** .
The letter on the grid for {11,6} is **D** . The minutes horse on move 19 is **L** .
The letter on the grid for {11,6} is **D** . The minutes horse on move 24 is **S** .

```
secondsLongitudeRaw16 = Total@(LetterNumber[#] & /@ Characters@"OAOEEEIV")
```

```
77
```

```
secondsLongitudeRaw6 = Total@(LetterNumber[#] & /@ Characters@"WDDILS")
```

```
71
```

Column 6 falls under sentence #24, already seen in the computation of minutes of latitude. It requires reversing the number obtained. So we take the extra **60 minutes to create a degree** and we are left with 11. Reversing this number does not change it. So we have 11 minutes.

Column 16 falls under sentence 39: "'Cause you never know what place you're gonna have to find." So we will **never know the place**. Why? Because it is to be removed. The contraction **'Cause** does remove the letters BE. The sum of these letters is 2+5=7. We apply such a contraction and remove the initial 7 in 77. We are left with 7 minutes.

The total is 11+7 = **18 minutes**

Degrees

Degrees are given by the collection of column #8 and #9

```
getColumnCoordinates[8]
```

```
<|15 → {1, 8}, 18 → {1, 8}|>
```

```
getColumnCoordinates[9]
```

```
⟨|2 → {11, 9}, 40 → {24, 9}|⟩
```

The letter on the grid for {1,8} is **M** . The degrees horse on move 15 is **O** .
The letter on the grid for {1,8} is **M**. The degrees horse on move 18 is **W** .

The letter on the grid for {11, 9} is **U** . The degrees horse on move 2 is **T** .
The letter on the grid for {24, 9} is **O** . The degrees horse on move 40 is **L** .

```
secondsLongitudeRaw = Total@(LetterNumber[#] & /@ Characters@"MMUOWTL") -
  Total@(LetterNumber[#] & /@ Characters@"O")
```

```
102
```

We add the one degree obtained in our computation of minutes, and get **103** degrees.

Then comes this very clever clue in sentence #1 : **"The town at the edge of the desert, like an oasis, had grown around springs of water that bubbled from the heart of the earth"**

Where is this desert in the heart of the earth?
It is in sentence #40: "I got all these degrees and meridians…south of **ea**st…no**rth** of west…any direc-t°n you want to go."

Shown in bold is the EARTH. And in its heart is the **ellipsis**. When these 3 dots "**bubble**", they trans-form into degree symbols. Note that the sentence tells us "I got all these degrees...". These **3** degrees need to be added to the longitude because we are talking about "...degrees and meridians..". Note also the reference to planets, typically on **elliptical** orbits.

So we have 103+3 = **106 degrees** .

LONGITUDE: 106°18'32"

THE HEADING

The above coordinates identify a small rectangle of forest near the 10th Mountain Division Memorial at Tennessee Pass, on Road 24 in Colorado. This rectangle is about 7800 sq.ft. We know (see video in the reference section) that the map arrows represent the configuration of trees and branches around the actual location of the famous X. But there is more.

There are two monuments at the memorial location. The smaller one has the date JUNE 11 1980 engraved at its base (see video). This is the father's grave. You can trace a vector (represented by an arrow) from the grave to the treasure location. To represent this vector, we need a length and a direction. Direction could be given with an angle (degrees) in reference to standard directions. Sentence #40 in Mr. Maps is an example of such directions: **"south of east…north of west"**.

It turns out that this information appears to be given by following our anagrams ODOMETER and WOOD TREE.

FOREST

With its trees, chapter 17 (FOREST) is the obvious candidate to provide this information. But before that, we need to have a quick look at the only odometer (or what looks like one) found in the book, in chapter 6 (FISH).

There are many possible interpretations of the whole FISH image, most involving the manipulation of bits, some using XOR, but here is a very simple one. Keeping with bits and bytes, we have 8 positions (byte) repeated twice (a **word**). And we have the letter **Y** repeated 6 times (the kimonos). So this reference to ODOMETER before going to WOOD TREE (FOREST) may mean that we have to count the number of letters (bits) in words containing the letter Y in FOREST. But, most importantly for that count, **our odometer is already set at 6**. So the result obtained in FOREST will add to this value. In fact, the letter **Y**, given its shape, is associated with a tree branching out.

The text of FOREST.

```
forestRaw =
  "Early the next morning, Amanda left the city behind. She traveled the back
    roads, staying in country inns and keeping country hours. She saw
    each thing with fresh eyes, and if it caught her interest, she noted
    it clearly in her book. The number of days devoted to her search no
    longer concerned Amanda. She accepted The Journey as her work. Each
    step was a step closer to Treasure. In this single-minded way, she
    moved forward. One day she came upon an extraordinary forest, with
    trees of strange and wondrous shapes. It was surrounded by a ring
    of silence and stone. And though the morning was clear, the forest
    inside was dense and mist. A tall stone archway marked the entrance.
    At its peak stood statues, like sentries on alert. Each held a musical
    instruments, and when Amanda approached, the statues creaked to life,
    dancing a mechanical minuet. The animals were charming, but like the
    trees, they were rather strange. Well, she realized, I have become a
    collector of strange things. And although the forest was obscured in
    the mist, she passed under the arch and entered. Now things became
    curiouser and curiouser. Amanda passed a lake whose surface reflected
    a castle. But when she looked up, she could see only clouds. Among the
    clouds sailed a big black hawk, which let out a harsch screech. Nothing
    else seemed to move. There were only trees and statues of people.
    From a distance, the statures seemed frozen in time, but their faces
    were pitted and worn with age. The inscriptions carved along their
    bases were chiseled in strange language she did not know. It is as if
    everything is in code, she thought. And I do not have the code book.
    Below one statue she found a book that might have helped. But it was
    written in the tongue of ancient Rome. So she copied only two short
    words and left the rest behind. Now the mist became a thick white fog.
    I'll never understand anything if I don't keep going, Amanda thought,
    even if I can't see where I am. Shapes soon appeared, and extraordinary
    shapes they were. There was a statue of a short man in a too-tall hat.
    Across from him was a rabbit, the handsomest she'd ever seen, well
    dressed in coat and vest. He held a watch with a fob. The watch was
    set precicely at eight. The fog thinned even more, to reveal several
    cats and a caterpillar. And hight above them all was the statue of a
    girl sitting on a mushroom. It was Alice from Wonderland. Why, she
    looks exactly like me, thought Amanda. It was a comforting thought.
    For the first time she could remember, she didn't feel alone. But
    she did feel tired, for there had been a great deal to see and even
    more to think about. I'll just rest for a minute, she told herself,
    as she joined Alice on the mushroom. On Alice's lap she put her
    book, and on the book she placed her head. Soon she began to dream.";
```

Sentences in FOREST, numbered.

```
forestSentences = TextSentences[forestRaw];
Column@
 Normal@(AssociationThread[Range[Length[forestSentences]], forestSentences])
```

```
1 → Early the next morning, Amanda left the city behind.
2 →
 She traveled the back roads, staying in country inns and keeping country hours.
3 → She saw each thing with fresh eyes, and if
   it caught her interest, she noted it clearly in her book.
4 → The number of days devoted to her search no longer concerned Amanda.
5 → She accepted The Journey as her work.
6 → Each step was a step closer to Treasure.
7 → In this single-minded way, she moved forward.
8 → One day she came upon an extraordinary
   forest, with trees of strange and wondrous shapes.
9 → It was surrounded by a ring of silence and stone.
10 → And though the morning was clear, the forest inside was dense and mist.
11 → A tall stone archway marked the entrance.
12 → At its peak stood statues, like sentries on alert.
13 → Each held a musical instruments, and when Amanda approached,
   the statues creaked to life, dancing a mechanical minuet.
14 → The animals were charming, but like the trees, they were rather strange.
15 → Well, she realized, I have become a collector of strange things.
16 → And although the forest was obscured
   in the mist, she passed under the arch and entered.
17 → Now things became curiouser and curiouser.
18 → Amanda passed a lake whose surface reflected a castle.
19 → But when she looked up, she could see only clouds.
20 → Among the clouds sailed a big black hawk, which let out a harsch screech.
21 → Nothing else seemed to move.
22 → There were only trees and statues of people.
23 → From a distance, the statures seemed frozen
   in time, but their faces were pitted and worn with age.
24 → The inscriptions carved along their
   bases were chiseled in strange language she did not know.
25 → It is as if everything is in code, she thought.
26 → And I do not have the code book.
27 → Below one statue she found a book that might have helped.
28 → But it was written in the tongue of ancient Rome.
29 → So she copied only two short words and left the rest behind.
30 → Now the mist became a thick white fog.
31 → I'll never understand anything if I don't
   keep going, Amanda thought, even if I can't see where I am.
32 → Shapes soon appeared, and extraordinary shapes they were.
33 → There was a statue of a short man in a too-tall hat.
```

```
34 → Across from him was a rabbit, the
     handsomest she'd ever seen, well dressed in coat and vest.
35 → He held a watch with a fob.
36 → The watch was set precicely at eight.
37 → The fog thinned even more, to reveal several cats and a caterpillar.
38 → And hight above them all was the statue of a girl sitting on a mushroom.
39 → It was Alice from Wonderland.
40 → Why, she looks exactly like me, thought Amanda.
41 → It was a comforting thought.
42 → For the first time she could remember, she didn't feel alone.
43 → But she did feel tired, for there
     had been a great deal to see and even more to think about.
44 → I'll just rest for a minute,
     she told herself, as she joined Alice on the mushroom.
45 → On Alice's lap she put her book, and on the book she placed her head.
46 → Soon she began to dream.
```

Words in FOREST.

```
forestWords = TextWords[forestRaw]
```

```
{Early, the, next, morning, Amanda, left, the, city, behind, She, traveled, the,
 back, roads, staying, in, country, inns, and, keeping, country, hours, She, saw,
 each, thing, with, fresh, eyes, and, if, it, caught, her, interest, she, noted,
 it, clearly, in, her, book, The, number, of, days, devoted, to, her, search, no,
 longer, concerned, Amanda, She, accepted, The, Journey, as, her, work, Each,
 step, was, a, step, closer, to, Treasure, In, this, single-minded, way, she,
 moved, forward, One, day, she, came, upon, an, extraordinary, forest, with,
 trees, of, strange, and, wondrous, shapes, It, was, surrounded, by, a, ring,
 of, silence, and, stone, And, though, the, morning, was, clear, the, forest,
 inside, was, dense, and, mist, A, tall, stone, archway, marked, the, entrance,
 At, its, peak, stood, statues, like, sentries, on, alert, Each, held, a, musical,
 instruments, and, when, Amanda, approached, the, statues, creaked, to, life,
 dancing, a, mechanical, minuet, The, animals, were, charming, but, like, the,
 trees, they, were, rather, strange, Well, she, realized, I, have, become, a,
 collector, of, strange, things, And, although, the, forest, was, obscured, in,
 the, mist, she, passed, under, the, arch, and, entered, Now, things, became,
 curiouser, and, curiouser, Amanda, passed, a, lake, whose, surface, reflected,
 a, castle, But, when, she, looked, up, she, could, see, only, clouds, Among,
 the, clouds, sailed, a, big, black, hawk, which, let, out, a, harsch, screech,
 Nothing, else, seemed, to, move, There, were, only, trees, and, statues, of,
 people, From, a, distance, the, statures, seemed, frozen, in, time, but,
 their, faces, were, pitted, and, worn, with, age, The, inscriptions, carved,
 along, their, bases, were, chiseled, in, strange, language, she, did, not,
 know, It, is, as, if, everything, is, in, code, she, thought, And, I, do, not,
 have, the, code, book, Below, one, statue, she, found, a, book, that, might,
 have, helped, But, it, was, written, in, the, tongue, of, ancient, Rome, So,
 she, copied, only, two, short, words, and, left, the, rest, behind, Now, the,
 mist, became, a, thick, white, fog, I'll, never, understand, anything, if,
 I, don't, keep, going, Amanda, thought, even, if, I, can't, see, where, I,
 am, Shapes, soon, appeared, and, extraordinary, shapes, they, were, There,
 was, a, statue, of, a, short, man, in, a, too-tall, hat, Across, from, him,
 was, a, rabbit, the, handsomest, she'd, ever, seen, well, dressed, in, coat,
 and, vest, He, held, a, watch, with, a, fob, The, watch, was, set, precicely,
 at, eight, The, fog, thinned, even, more, to, reveal, several, cats, and, a,
 caterpillar, And, hight, above, them, all, was, the, statue, of, a, girl,
 sitting, on, a, mushroom, It, was, Alice, from, Wonderland, Why, she, looks,
 exactly, like, me, thought, Amanda, It, was, a, comforting, thought, For,
 the, first, time, she, could, remember, she, didn't, feel, alone, But, she,
 did, feel, tired, for, there, had, been, a, great, deal, to, see, and, even,
 more, to, think, about, I'll, just, rest, for, a, minute, she, told, herself,
 as, she, joined, Alice, on, the, mushroom, On, Alice's, lap, she, put, her,
 book, and, on, the, book, she, placed, her, head, Soon, she, began, to, dream}
```

Words in FOREST containing the letter Y

```
yWords = If[MemberQ[Characters@#, "Y"], #, Nothing] & /@ ToUpperCase@forestWords
```

```
{EARLY, CITY, STAYING, COUNTRY, COUNTRY, EYES, CLEARLY, DAYS, JOURNEY,
 WAY, DAY, EXTRAORDINARY, BY, ARCHWAY, THEY, ONLY, ONLY, EVERYTHING,
 ONLY, ANYTHING, EXTRAORDINARY, THEY, PRECICELY, WHY, EXACTLY}
```

```
StringLength[StringJoin@yWords]
```

```
150
```

There may be some special rules to apply to some of the words. For examples, does EXTRAORDINARY means that we should add an extra word ORDINARY? The word appears twice, which would increase our count from 150 to 152.

We will use 150, to which we add 6 already on the ODOMETER. So we have **156 meters** (after all , it is an odo**meter**).

In sentences #27 and #28, we are told that a book written in the tongue of ancient Rome may help. The tongue of ancient Rome is Latin, an **italic** language. We conclude that what is in italic may help.
Here are some sentences where we find italics.

- Sentence #15 tells us that we have to collect something. We have collected Y.
- Sentence #40 has the word “Why” (Y)
- Sentences #25 and #26 are the most important, telling us that we need the **code book**.
- Sentence #31 suggests that we have to find out **where we are**
- We will come back to #40 and #44.

It turns out that we have to find out precisely where we are. What are the coordinates of this forest? The answer is in the code book, or more precisely in the code assigned to books: the **ISBN**.
The 10-digits ISBN for this book is **0446381608**.

Sentence #29 tells us to copy two short words and leave the rest behind. A short word is 16 bits, two short words is 32 bits. We take 32 out of the ISBN. This requires taking 638168 (sum of these numbers is 32). These make up a coordinate: 63 degrees 81 minutes 68 seconds. Converting excess seconds to minutes and excess minutes to degrees we get a longitude of **64° 22’ 08”** .

Now lets look at the rest, what was left behind: 0448. This is a coordinate with the degrees missing. Sentence #44 talks about “just rest for a minute”. This is because the degrees are equal to 0. We have a latitude of **00° 04’ 08”** . We are in the **Amazon Rain Forest**, this is why it is so misty! Sentence #40 about “she looks exactly like me” is a way to say “equate her” (EQUATOR).

Here, the longitude is important. Sentence #2 talks about “keeping country hours”. Hours in a country (time zones) are determined by the longitude. We need to **keep** the hours means that we only keep the **64 degrees**.

In summary, starting at the grave (small monument) the treasure is located **156 meters away at an angle of 64 degrees**. This happens to point inside the small rectangle delimited by our latitude and longitude.

FINAL LOCATION

These images from Google Earth give us an approximate view.
The top right corner image is a view of the Tennessee Pass Memorial from road 24. In a triangular area we see a flag pole, the big monument, and the small monument (father's grave).
On the aerial view, we see clearly the flag pole and the large monument, but the small monument is hidden behind trees.
The yellow rectangle delimits the area falling within our longitude and latitude.
The red line has a length of 156 meters, starting at the small monument (guessing the exact position) at an angle of 64 degrees East of South and ending in the yellow rectangle, likely the treasure location.

END OF JOURNEY