

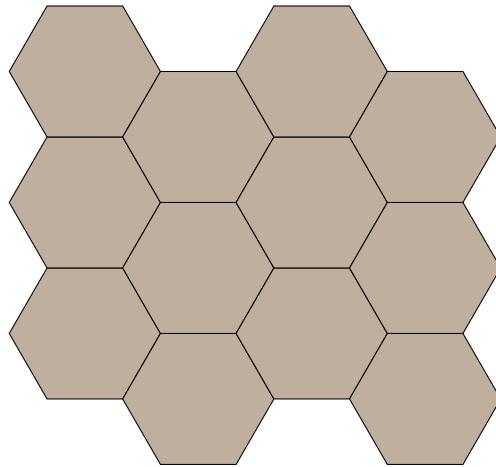
E. 芽芽農場遊戲

Description

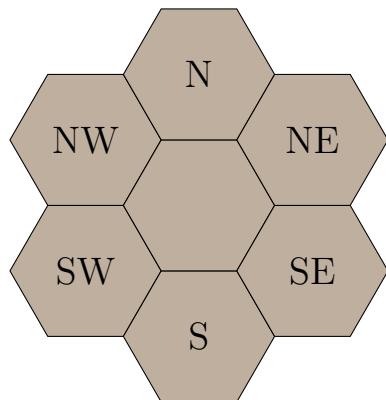
芽芽農場遊戲是一款操作簡單，卻容易上癮的遊戲。遊戲的規則如下：

遊戲的地圖由若干個正六邊形的格子所組成，總共有 m 個直排緊緊連接著彼此，每一排有 n 個六邊形。如果每一直排都由上到下對六邊形依序編號為 $1, 2, \dots, n$ ，那奇數排的第 i 個六邊形會跟左右兩排的第 $i - 1$ 與第 i 個六邊形相鄰（反過來說，偶數排的第 i 個六邊形會與左右兩排的第 i 與第 $i + 1$ 個六邊形相鄰）。

以下是一個 $n = 3$ 、 $m = 4$ 的空白地圖例子。

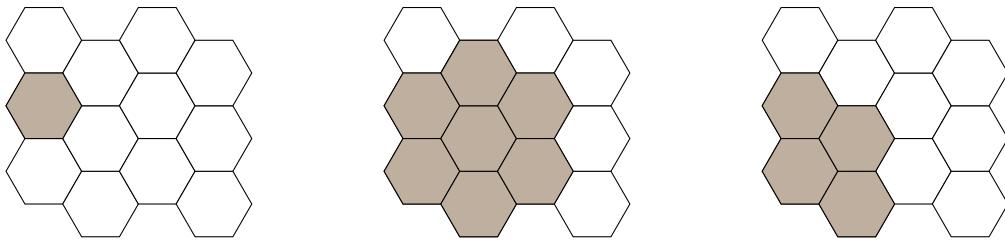


整張地圖除了邊界的部份，每個格子都剛剛好有六個鄰居，我們用北、東北、西北、南、東南、西南來表示一個格子的所有鄰居，如下圖所示。遊戲的主角一步可以移動到某個相鄰的格子。



地圖有三種地形：平原、海洋與山地。因為遊戲的主角芽芽會爬山涉水，所以只要不要超出邊界就可以通行。

另外，在平原上可以興建農場，也就是這個遊戲的主角。一個農場只能全部都蓋在平原上，不可以超出邊界外，而且必須蓋成完美對稱的樣子：一個農場會由一個半徑 r 跟一個中心決定，這樣所決定的農場範圍是「從中心開始，走 r 步以內可以走到的所有格子」，這裡的 r 可以是任意的非負整數，也就是說他可以是 $0, 1, 2, \dots$ 的數字。例如說，可以蓋成自己一個格子（這時候半徑 $r = 0$ ）或是一個格子與他的所有鄰居（這時候半徑 $r = 1$ ），但是一個像是平行四邊形的格子們就不可能由單一一個農場所組成。



這些農場在興建的時候必須要有這樣的條件，不過在這個遊戲內可以興建很多農場，所以說玩家可以蓋兩個「只有一個格子的農場」相接著彼此，乍看之下好像不是合法的方法，但是只要兩個農場是分開興建的，就是遊戲內允許的。

遊戲還額外提供了玩家在探索地形中的指示，總共有 26 個標示可以使用，不過每個標記同時都只能存在一個，若是已經使用過的標記再度被使用一次，那之前的標記就會被撤銷。一塊地同時也只能有一個標記，如果標記前，格子上已經有另一個標記，格子上原本的標記會先被移除，再打上新的標記。

芽芽太喜歡這款遊戲了，因此想要用終端機的文字輸出復刻這款遊戲！你能幫幫芽芽嗎？

Input

輸入的第一行有兩個以空白分開的整數 n, m ，代表遊戲的高度與寬度。

接下來有 $2n$ 行，每兩行為一個單位，第 i 個單位（也就是第 $2i - 1, 2i$ 行）輸入的是所有直排的第 i 個格子的地形。字元「.」、「~」、「^」，分別代表平原、海洋與山地。農場不會出現在輸入的地圖中。其中，奇數行的地形會在第一行輸入、偶數行的地形會在第二行輸入。另外，奇數行的兩個地形之間會由三個空白作為間隔，而偶數行的兩個地形之間也會由三個空白作為間隔，而且偶數行的前綴會有兩個空白。

再接下來的一行有兩個以空白分開的整數 r, c ，代表玩家一開始在第 c 直排的第 r 個格子上。

最後，遊戲開始。接下來會有數行，每一行是一個遊戲操作，格式必定是下面數個其中之一：

- move d ：移動。
 - d 一定是 N、NE、NW、S、SE、SW 其中之一，依序表示玩家想要往北、東北、西北、南、東南、西南的方向移動一格。

- 保證玩家不會移動到邊界之外。
 - `mark c`：標記。
 - c 一定是英文大寫字母之一，表示要使用的標記。
 - 如果標記已經被使用過，需要額外將地圖上前一次同樣的標記清除。
 - `build`：建造農場。
 - 在玩家當下的格子興建一個半徑盡量大的農場。
 - 注意農場必須都在邊界以內！
 - 保證玩家只會在沒有興建農場的平原地形上做這個操作。
 - `demolish`：拆除農場。
 - 在玩家當下格子的農場整個拆除。
 - 如果當初興建的農場是多於一格的農場，則需全數拆除。
 - 保證玩家只會在農場內做這個操作。
 - `finish h w`：結束遊戲。
 - 保證最後一個操作必定是這個操作，而且這個操作只會出現在最後一個操作。
 - 輸出格式由下一段落所述。
-
- $2 \leq n, m \leq 20$ 。
 - $1 \leq r \leq n$ 。
 - $1 \leq c \leq m$ 。
 - $1 \leq h, w \leq 201$ ， h, w 都是奇數。
 - 操作數量 ≤ 100 。

Output

對於 `finish` 操作的輸出，請按照下面的格式：

首先，一個格子由下面的方式表示：

```
*---*  
/AAAAA\  
*AAABAAA*  
\AAAAA/  
*---*
```

其中，

- 如果這個格子是某個農場的一部分，則 A 需要被替換為 #，否則，如果地形是平原、山地、海洋，則 A 需要被分別替換為「_」（空白）、「~」以及「~」。
- 如果這個格子有被標記，則 B 需要被替換成標記對應的字母。否則 B 與 A 相同。

我們稱 B 為一個格子的中心點。

你需要輸出 $h + 2$ 列，每列有 $w + 2$ 個字元（不含最後的換行字元）。

- 對於第一列與最後一列，請依序輸出一個 +、 w 個 -、與再一個 +。
- 對於中間的其他列，第一個與最後一個字元是一個 |。
- 剩下的 $h \times w$ 區域，我們稱為繪畫區域。

在繪畫區域中，玩家所在的格子的中心必須在繪畫區的中心（也就是第 $\frac{h+1}{2}$ 列、第 $\frac{w+1}{2}$ 個字元）。其他格子的中心位置由他相對於已經決定位置的鄰居，由下面的規則決定：

- 它是北方鄰居：中心位置在它鄰居的中心位置往上 4 列。
- 它是東北方鄰居：中心位置在它鄰居的中心位置往上 2 列、往右 6 個字元。
- 它是西北方鄰居：中心位置在它鄰居的中心位置往上 2 列、往左 6 個字元。
- 它是南方鄰居：中心位置在它鄰居的中心位置往下 4 列。
- 它是東南方鄰居：中心位置在它鄰居的中心位置往下 2 列、往右 6 個字元。
- 它是西南方鄰居：中心位置在它鄰居的中心位置往下 2 列、往左 6 個字元。

這樣只需要確定玩家所在的格子的中心就可以確定所有格子的中心位置。

每一行的結尾都需要輸出一個換行字元。超出繪畫區域的部份不需要也不應該輸出。

請注意，本題採用**嚴格比對**，任何空白或換行的錯誤都會被算成輸出不正確。你可以在系統上下載包含所有範例測試資料的壓縮檔檢查實際的範例輸出。

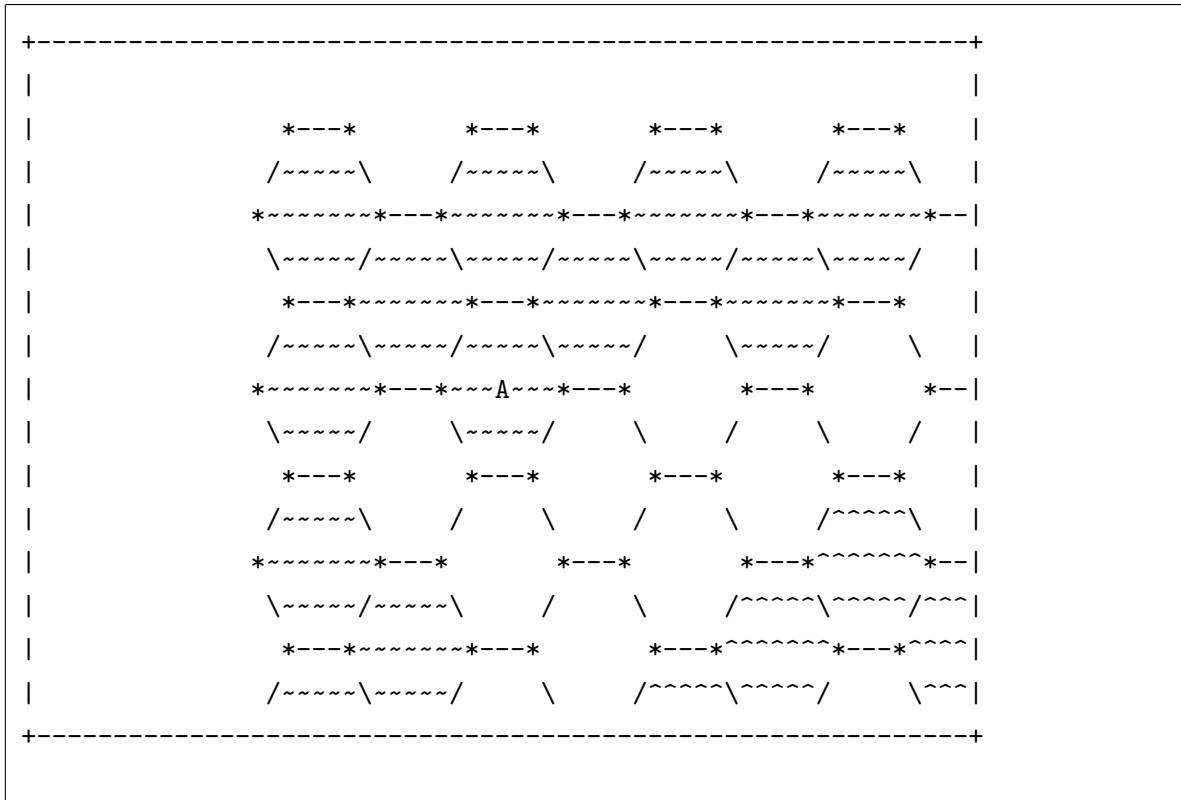
Sample 1

Input
2 4
~ ~
~ ~
~ ~
.
1 3
finish 7 61
Output

```
+-----+
|           *---*      *---*
|          /~~~~~\     /~~~~~\
| *~~~~~*---*~~~~~*---*
| \~~~~~/~~~~~\~~~~~/~~~~~\
| *---*~~~~~*---*~~~~~*
| /~~~~~\~~~~~/~~~~~\~~~~~/
+-----+
```

Sample 2

Input
4 9
~ ~ ~ ~ ~
~ ~ ~ .
~ ~ . . .
. . . .
~ . ^ ^
~ . ^ ^
~ . ^ . .
~ . . .
2 3
mark A
finish 15 61
Output



Sample 3

Input	Output
<pre>4 9 ~ ~ ~ ~ ~ ~ ~ ~ . ~ ~ ~ . ^ ^ ~ . ^ ^ ~ . . 2 6 move S move SW finish 1 1</pre>	<pre>+++ ^ +++</pre>

Sample 4

Input	Output
4 9 ~ ~ ~ ~ ~ ~ ~ ~ . . ~ ~ ~ . ^ ^ ~ ~ . ^ ^ ~ ~ . ^ . . ~ 2 6 move S mark A move NE mark B move SW finish 3 3	+---+ ^ ^ ^ A ^ ^ ^ +---+

Sample 5

Input
4 9 ~ ~ ~ ~ ~ ~ ~ ~ . . ~ ~ ~ . ^ ^ ~ ~ . ^ ^ ~ ~ . ^ . . ~ 3 5 build finish 21 61
Output

```
+-----+
|      *---*      *---*      *---*      *---*      *---*      |
|      /~~~~~\      /~~~~~\      /~~~~~\      /~~~~~\      /~~~~~\      |
|  *~~~~~*---*~~~~~*---*~~~~~*---*~~~~~*---*~~~~~*---*~~~~~*   |
|  \~~~~~/~~~~~\~~~~~/~~~~~\~~~~~/~~~~~\~~~~~/~~~~~\~~~~~/~~~~~\   |
|  *---*~~~~~*---*~~~~~*---*~~~~~*---*~~~~~*---*~~~~~*---*~~~~~*   |
|  /~~~~~\~~~~~/~~~~~\~~~~~/#####\~~~~~/~~~~~\~~~~~/~~~~~\~~~~~/~~~~~\   |
|  *~~~~~*---*~~~~~*---*#####*---*#####*---*~~~~~*---*~~~~~*---*~~~~~*   |
|  \~~~~~/~~~~~/#####\#####/#####\#####\~~~~~/~~~~~/#####\#####\~~~~~/~~~~~\   |
|  *---*      *---*#####*---*#####*---*#####*---*~~~~~*---*~~~~~*---*~~~~~*   |
|  /~~~~~\~~~~~/~~~~~/#####\#####\#####/~~~~~/~~~~~/#####\#####\~~~~~/~~~~~\   |
|  *~~~~~*---*      *---*#####*---*~~~~~*---*~~~~~*---*~~~~~*---*~~~~~*   |
|  \~~~~~/~~~~~/~~~~~/#####\~~~~~/~~~~~/#####\~~~~~/~~~~~/#####\~~~~~/~~~~~\   |
|  *---*~~~~~*---*      *---*      *---*      *---*      *---*      |
|      \~~~~~/~~~~~/~~~~~/~~~~~/~~~~~/~~~~~/~~~~~/~~~~~/~~~~~/~~~~~/~~~~~/~~~~~\   |
|  *---*      *---*      *---*      *---*      *---*      |
+-----+
```

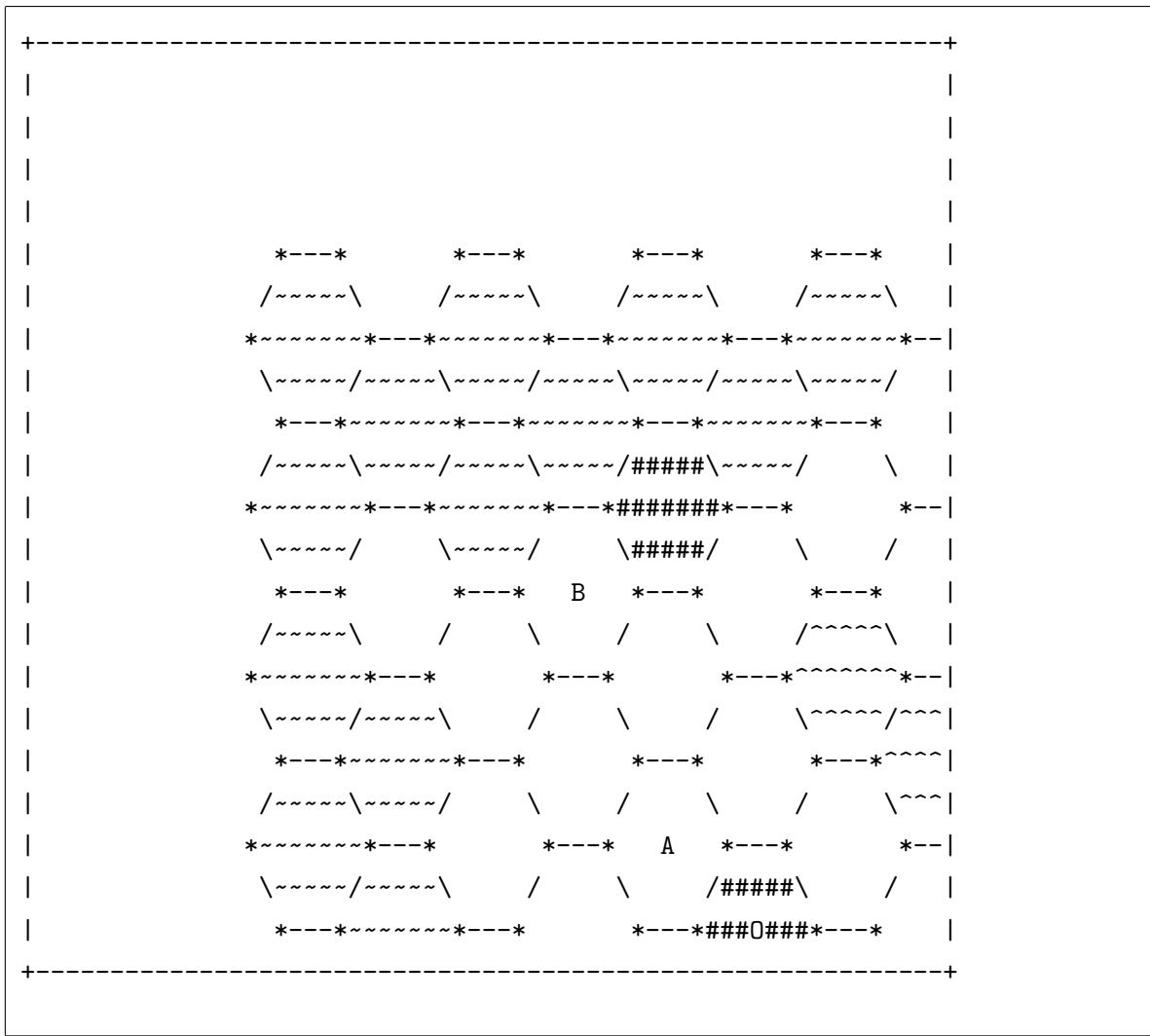
Sample 6

Input

```
4 9
~ ~ ~ ~ ~
~ ~ ~ .
~ ~ . .
.
.
~ . . ^ ^
~ . .
~ . .
~ . .

3 5
move N
build
mark O
move S
move S
mark A
move SE
mark O
build
move N
move NW
mark B
move SW
build
move N
mark B
demolish
move NW
finish 21 61
```

Output



配分

在一個子任務的「測試資料範圍」的敘述中，如果存在沒有提到範圍的變數，則此變數的範圍為 Input 所描述的範圍。

子任務編號	子任務配分	測試資料範圍
1	0%	範例測試資料
2	15%	只有 move 與 finish 操作，且 $h = w = 1$
3	15%	只有 move、mark 與 finish 操作，且 $h = w = 3$
4	15%	只有 finish 操作
5	15%	只有 move 與 finish 操作
6	15%	只有 move、mark 與 finish 操作
7	15%	只有 move、mark、build、finish 操作
8	10%	無額外限制

Hint

下圖的座標系可能會給你很大的幫助，不過這是為什麼呢？

