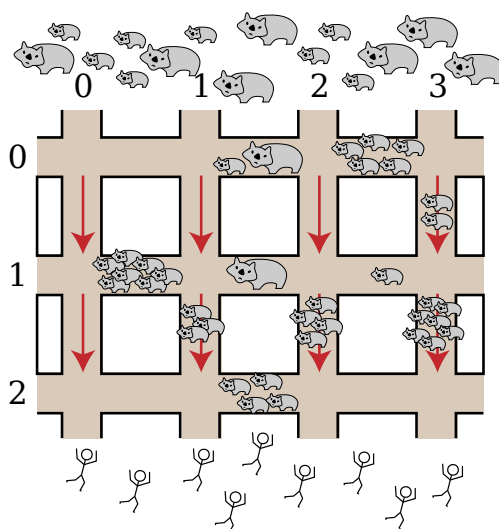


חתולים אוסטרליים (Wombats) מתקיפים את העיר בריסביין, ועליכם להוביל את אנשי העיר למקום מבטחים.

הכבישים (Roads) בבריסביין בנויים בצורת רשת (Grid). יש R כבישים אופקיים (מזרח-מערב) שממוספרים $0, \dots, (R - 1)$ לפי הסדר מצפון לדרום. יש C כבישים אנכיים (צפון-דרום), שממוספרים $0, \dots, (C - 1)$ לפי הסדר ממערב למזרח. ראו דוגמא באיור הבא.

(Horizontal = אופקי) (Vertical = אנכי)



החתולים פולשים מכיוון צפון והאנשים בורחים לכיוון דרום. על דרכים אופקיות, כל איש יכול לנוע בשני הכיוונים. אבל, על דרכים אנכיות, כל איש יכול לנוע רק בכיוון דרום, לכיוון מקום מבטחים.

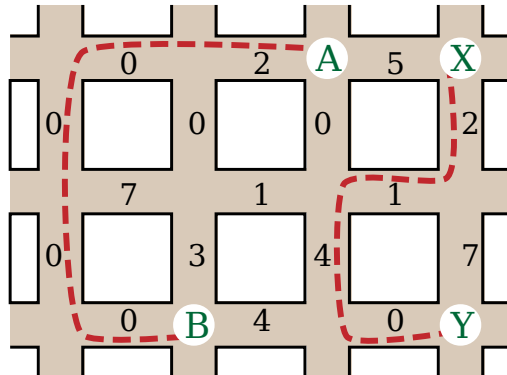
כל כביש אופקי P וכביש אנכי Q מגדירים ביחד צומת (Intersection) המסומן (P, Q) . בכל קטע כביש המחבר שני צמתים סמוכים, יש מספר מסויים של חתולים. מספר החתולים בקטע כביש יכול להשתנות עם הזמן. אתם תקבלו משימות, שבכל אחת מהן תתבקשו להוביל אדם אחד מצומת נתון בכביש הצפוני (כביש אופקי מספר 0) לצומת נתון בכביש הדרומי (כביש אופקי מספר $R-1$). מטרתכם היא שהמסלול יעבור דרך מספר כולל מינימלי של חתולים.

בהתחלה, נתונים מימדי הרשת (Grid) ולכל קטע כביש המחבר בין צמתים סמוכים, נתון מספר החתולים בו. לאחר-מכן, תקבלו סדרה של E מאורעות בהם תצטרכו לטפל. כל מאורע הוא מאחד משני הסוגים הבאים:

- מאורע *change* - מאורע שמשנה את מספר החתולים על קטע כביש מסויים.
- מאורע *escape* - מאורע שבו אתם מתבקשים להוביל אדם אחד מצומת נתון בכביש אופקי מספר 0 לצומת נתון בכביש אופקי מספר $R-1$. על המסלול לעבור דרך מספר כולל מינימלי של חתולים.

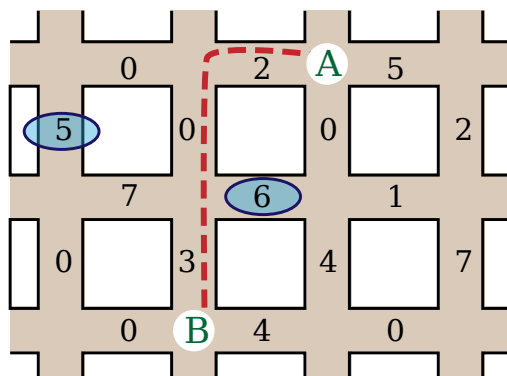
You must handle these events by implementing the routines `init()`, `changeH()`, `changeV()` and `escape()`, as described below.

דוגמא



באיור למעלה מתואר מצב התחלתי עם $R=3$ כבישים אופקיים ו- $C=4$ כבישים אנכיים. מספר החתולים על כל קטע כביש מצוין באיור. הביטו בסדרת המאורעות הבאה:

- מאורע `escape` שבו אתם נדרשים להוביל אדם מהצומת $A = (0, 2)$ אל הצומת $B = (2, 1)$. המסלול האופטימלי מתואר באיור למעלה והוא עובר דרך 2 חתולים.
- מאורע `escape` שבו אתם נדרשים להוביל אדם מהצומת $X = (0, 3)$ אל הצומת $Y = (2, 3)$. המסלול האופטימלי מתואר גם הוא באיור למעלה והוא עובר דרך 7 חתולים.
- לאחר מכן מתרחשים שני מאורעות `change`: מספר החתולים בקטע הצפוני ביותר בכביש אנכי מספר 0 משתנה ל- 5. מספר החתולים בקטע האמצעי בכביש אופקי מספר 1 משתנה ל- 6. שינויים אלו מודגשים באיור למטה באמצעות עיגולים כחולים.



- מאורע `escape` נוסף שבו אתם נדרשים להוביל אדם מהצומת $A = (0, 2)$ אל הצומת $B = (2, 1)$. המסלול האופטימלי מתואר באיור למטה והוא עובר דרך 5 חתולים.

Implementation

You should submit a file implementing the procedures `init()`, `changeH()` and `changeV()` and the function `escape()`, as follows:

Your Procedure: `init()`

C/C++ `void init(int R, int C, int H[5000][200], int V[5000][200]);`

Pascal `type wombatsArrayType = array[0..4999, 0..199] of LongInt;
procedure init(R, C : LongInt; var H, V : wombatsArrayType);`

Description

בפונקציה הזו נתון לכם המצב ההתחלתי. אתם יכולים לאתחל משתנים גלובליים ומבני נתונים. מערכת הבדיקה תקרא לפונקציה הזו פעם אחת בדיוק, בהתחלה. רק לאחר מכן היא תקרא לפונקציות האחרות.

Parameters

- `R` : מספר הכבישים האופקיים.
- `C` : מספר הכבישים האנכיים.
- `H` : מערך דו-ממדי בגודל $R \times (C - 1)$ שבו התא `H[P][Q]` מתאר את מספר החתולים על קטע הכביש האופקי המחבר בין הצמתים $(P, Q) - 1$ ו- $(P, Q + 1)$.
- `V` : מערך דו-ממדי בגודל $(R - 1) \times C$ שבו התא `V[P][Q]` מתאר את מספר החתולים על קטע הכביש האנכי המחבר בין הצמתים $(P, Q) - 1$ ו- $(P + 1, Q)$.

Your Procedure: `changeH()`

C/C++ `void changeH(int P, int Q, int W);`

Pascal `procedure changeH(P, Q, W: LongInt);`

Description

הפונקציה הזו משנה את מספר החתולים בקטע הכביש האופקי שבין הצמתים (P, Q) ו- $(P, Q + 1)$.

Parameters

- P : מציין איזה כביש אופקי מושפע מהשינוי (מתקיים $0 \leq P \leq R - 1$).
- Q : מציין בין אילו שני כבישים אנכיים נמצא קטע הכביש (מתקיים $0 \leq Q \leq C - 2$).
- W : מספר החתולים החדש בקטע הכביש הזה (מתקיים $0 \leq W \leq 1,000$).

Your Procedure: `changeV()`

C/C++ `void changeV(int P, int Q, int W);`

Pascal `procedure changeV(P, Q, W: LongInt);`

Description

הפונקציה הזו משנה את מספר החתולים בקטע הכביש האנכי המחבר בין הצמתים (P, Q) - $(P + 1, Q)$.

Parameters

- P : מציין בין אילו שני כבישים אופקיים נמצא קטע הכביש (מתקיים $0 \leq P \leq R - 2$).
- Q : מציין איזה כביש אופקי מושפע מהשינוי (מתקיים $0 \leq Q \leq C - 1$).
- W : מספר החתולים החדש בקטע הכביש הזה (מתקיים $0 \leq W \leq 1,000$).

Your Function: `escape()`

C/C++ `int escape(int V1, int V2);`

Pascal `function escape(V1, V2 : LongInt) : LongInt;`

Description

הפונקציה הזו צריכה לחשב את מספר החתולים המינימלי האפשרי במסלול מהצומת $(0, V1)$ אל הצומת $(R-1, V2)$.

Parameters

- $V1$: מציין איפה בכביש אופקי 0 מתחיל המסלול (מתקיים $0 \leq v1 \leq c-1$).
- $V2$: מציין איפה בכביש אופקי R-1 מסתיים המסלול (מתקיים $0 \leq v2 \leq c-1$).
- הפונקציה צריכה להחזיר את מספר החתולים במסלול אופטימלי.

Sample Session

The following session describes the example above:

Function Call	Returns
<code>init(3, 4, [[0,2,5], [7,1,1], [0,4,0]], [[0,0,0,2], [0,3,4,7]])</code>	
<code>escape(2,1)</code>	2
<code>escape(3,3)</code>	7
<code>changeV(0,0,5)</code>	
<code>changeH(1,1,6)</code>	
<code>escape(2,1)</code>	5

Constraints

- מגבלת זמן: 20 שניות
- מגבלת זכרון: 256MiB, כלומר, 256 כפול 1048576 בתים
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- יש לכל היותר 500 מאורעות `change` (כלומר, קריאות ל- `changeH()` או `changeV()`)
- יש לכל היותר 200,000 קריאות ל- `escape()`
- בכל זמן נתון, יש לכל היותר 1,000 חתולים בכל קטע כביש

Subtasks

Subtask	Points	Additional Input Constraints
1	9	$C = 1$
2	12	$R, C \leq 20$, and there will be no calls to <code>changeH()</code> or <code>changeV()</code>
3	16	$R, C \leq 100$, and there will be at most 100 calls to <code>escape()</code>
4	18	$C = 2$
5	21	$C \leq 100$
6	24	(None)

Experimentation

The sample grader on your computer will read input from the file `wombats.in`, which must be in the following format:

- line 1: `R C`
- line 2: `H[0][0] ... H[0][C-2]`
- ...
- line $(R + 1)$: `H[R-1][0] ... H[R-1][C-2]`
- line $(R + 2)$: `V[0][0] ... V[0][C-1]`
- ...
- line $(2R)$: `V[R-2][0] ... V[R-2][C-1]`
- next line: `E`
- next `E` lines: one event per line, in the order in which events occur

If $C = 1$, the empty lines containing the number of wombats on horizontal roads (lines 2 through $R + 1$) are not necessary.

The line for each event must be in one of the following formats:

- to indicate `changeH(P, Q, W)`: `1 P Q W`
- to indicate `changeV(P, Q, W)`: `2 P Q W`
- to indicate `escape(V1, V2)`: `3 V1 V2`

For instance, the example above should be provided in the following format:

```
3 4
0 2 5
7 1 1
0 4 0
0 0 0 2
0 3 4 7
5
3 2 1
3 3 3
2 0 0 5
1 1 1 6
3 2 1
```

Language Notes

C/C++ You must `#include "wombats.h"`.

Pascal You must define the `unit Wombats`. All arrays are numbered beginning at `0` (not `1`).

See the solution templates on your machine for examples.