Hong Kong Olympiad in Informatics 2020/21 Senior Group

Task Overview

ID	Name	Time Limit	Memory Limit	Subtasks
S211	Skyscraperhenge	1.000 s	256 MB	21 + 19 + 29 + 31
S212	Super Chat	1.000 s	256 MB	16 + 15 + 18 + 27 + 24
S213	Chinese Checkers	1.000 s	256 MB	2 + 3 + 4 + 5 + 6 + 14 + 15 + 19 + 32

Notice:
Unless otherwise specified, inputs and outputs shall follow the format below:

- One space between a number and another number or character in the same line.
- No space between characters in the same line.
- Each string shall be placed in its own separate line.
- Outputs will be automatically fixed as follows: Trailing spaces in each line will be removed and an end-of-line character will be added to the end of the output if not present. All other format errors will not be fixed.

C++ programmers should be aware that using C++ streams (`cin` / `cout`) may lead to I/O bottlenecks and substantially lower performance.

For some problems 64-bit integers may be required. In Pascal it is `int64`. In C/C++ it is `long long` and its token for `scanf`/`printf` is `%lld`.

All tasks are divided into subtasks. You need to pass all test cases in a subtask to get points.

S211 - SKYSCRAPERHENGE

Time Limit: 1.000 s / Memory Limit: 256 MB

In Byteland, there is a special residential area called "Skyscraperhenge", where some buildings are arranged in a circle.

There are N building zones in Skyscraperhenge, numbered from 0 to $N-1$ in clockwise order. Zone $i+1$ is to the right of zone i for $0 \le i < N-1$, and zone 0 is to the right of zone $N-1$. On top of each zone contains a building numbered in correspondence with the zone. The height of building i is represented by an integer A_i.

One day, you decide to test out a drone in Skyscraperhenge. At time 0, the drone is located in zone 0, at height $H+0.5$ above ground ($H \ge A_0$).

A series of control commands is given as a string S of length K. Each character of the string is either `D` or `N`, representing the two movements of the drone:

- `D`: The drone is lowered by 1 unit, i.e. for the drone located at height y, it will descend to height $y-1$.
- `N`: The drone fly from its current zone to the next zone, with its height remain the same, i.e. for the drone located at zone x, if $0 \le x < N-1$, it will move to zone $x+1$; else, if $x = N-1$, it will move to the zone 0.

The drone would move according to the string of commands repeatedly. Suppose S_i represents the i^{th} character of the string. At first, the drone would execute the command represented by S_1, then that of S_2, S_3, etc. After finishing the command S_K, the movement process will be repeated by executing command S_1 again. For example, is $S =$ `DNND`, the drone would go down `D` $\rightarrow$ go to the next zone `N` $\rightarrow$ go to the next zone `N` $\rightarrow$ go down `D` $\rightarrow$ go down `D` $\rightarrow$ go to the next zone `N` $\rightarrow$ go to the next zone `N` $\rightarrow$ go down `D`... However, at any point of time, if the drone crashes into any building, it will stop immediately.

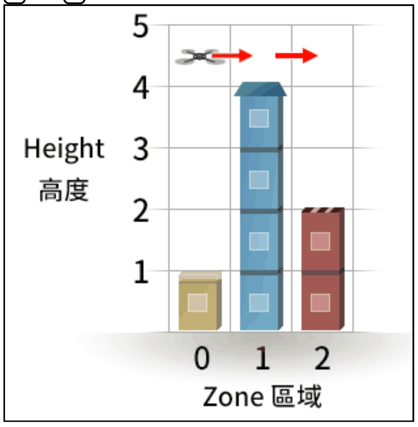
Mathematically, the drone crashing process is defined as follows:

- For a drone in zone x descending into height $y-1$ from height y, and $y-1 < A_x$, it will crash into building x on the **top**.
- For a drone flying into zone x with height y, and $y < A_x$, it will crash into building x on the **side**.

The following scenarios are examples of the drone crashing process, all of them started with the drone placed in zone 0, with height 4.5:

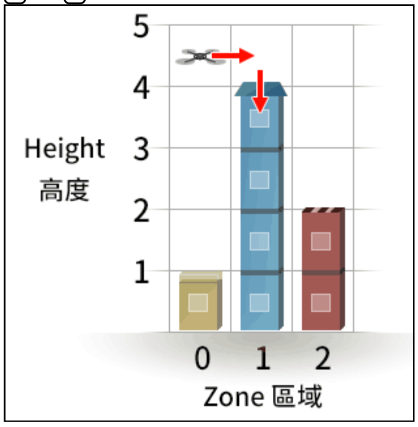
No crash:

`N` $\rightarrow$ `N`



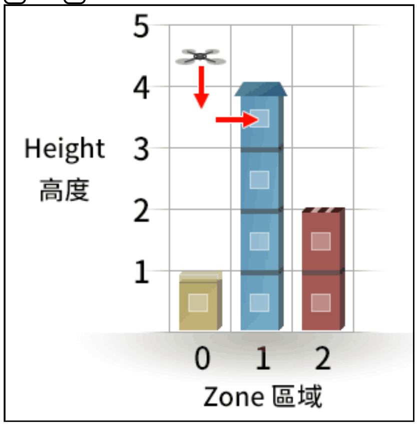
Crashes into building 1 on the top:

`N` $\rightarrow$ `D`



Crashes into building 1 on the side:

`D` $\rightarrow$ `N`



(Noted that the figures only described the flat view of Skycraperhenge. The buildings are arranged in a circle.)

Your job is to determine whether the drone will fly forever or not, and if not, determine which building it crashes into and how it happens.

INPUT

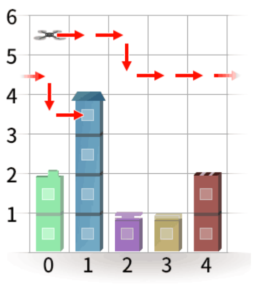
The first line contains two integers, N and H. N is the number of building zones and $H + 0.5$ is the initial height of the drone.
The second line contains N integers, $A_0, A_1, A_2, \ldots, A_{N-1}$, the heights of the buildings. It is guaranteed that the initial height of the drone is higher than the height of building 0, i.e. $H \geq A_0$.
The third line contains an integer K, the length of the command sequence.
The fourth line contains a string S of length K, the sequence of commands that the drone follows. The string contains only `D` and `N`.

OUTPUT

If the drone flies forever without stopping, output `FOREVER` in the first and only line.
If the drone crashes into building i on the top and stop, output `TOP` in the first line and an integer i in the second line.
If the drone crashes into building i on the side and stop, output `SIDE` in the first line and an integer i in the second line.

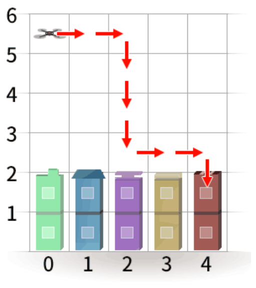
SAMPLE TESTS

	Input	Output
1	5 5 2 4 1 1 2 4 NNDN	SIDE 1



	Input	Output
2	5 5 2 2 2 2 2 5 NNDDD	TOP 4

This sample satisfies the constraints of Subtask 2.



	Input	Output
3	3 2 2 2 2 1 N	FOREVER
4	5 10 1 8 9 8 9 2 DD	TOP 0

SUBTASKS

For all cases:
$3 \leq N \leq 100000$
$A_0 \leq H \leq 2 \times 10^9$
$1 \leq A_i \leq 2 \times 10^9$
$1 \leq K \leq 100000$

	Points	Constraints
1	21	$3 \leq N \leq 100$ $A_0 \leq H \leq 100$ $1 \leq A_i \leq 100$ $1 \leq K \leq 100$ It is guaranteed that the drone will crash and stop at some point
2	19	All A_i are equal In string S, no command $\boxed{\texttt{N}}$ appears after a command $\boxed{\texttt{D}}$
3	29	In string S, at least half of the commands are $\boxed{\texttt{N}}$
4	31	No additional constraints

S212 - SUPER CHAT

Time Limit: 1.000 s / Memory Limit: 256 MB

Percy is a VTuber (Virtual YouTuber) who regularly does live streams. Percy's fans can communicate with him during the stream using the chat functionality next to the video player. Since Percy has a lot of fans, most chat messages go unnoticed. A great way to get the message noticed by the streamer is by purchasing Super Chats. (Figure 2)

Once a user pays for a Super Chat, a button that can reveal the chat message will be pinned at the top of the chat window for a certain continuous duration. Once the pin duration has passed, the Super Chat expires and is removed from the top of the chat window. The duration is determined by the price chosen by the user. The following table shows the pricing tiers and their pin durations.

Price	Colour	Pin duration and notes
5 – 9	Blue	0 minutes. No chat message can be entered.
10 – 24	Cyan	0 minutes
25 – 49	Green	2 minutes
50 – 99	Yellow	5 minutes
100 – 249	Orange	10 minutes
250 – 499	Magenta	30 minutes
500 – 999	Red	1 hour
1000 – 1499	Red	2 hours
1500 – 1999	Red	3 hours
2000 – 2499	Red	4 hours
$2500	Red	5 hours

Notes: The minimum price that can be chosen by the user is 5 and the maximum is 2500.
Blue and Cyan Super Chats will not be pinned at the top of the chat window.

Percy is so famous that he receives a lot of Super Chats. The Super Chat section of the chat window can only display at most 3 Super Chats at a time. When there are more than 3 pinned Super Chats, the 3 latest pinned Super Chats ordered by purchase time will be shown. Those older than the 3rd one (according to purchase time) are not visible (Figure 3). It is possible for a Super Chat to become invisible and then visible again if, for example, it is the 4th newest pinned Super Chat and one of the 3 newer Super Chat expires.

Percy has just ended the stream after streaming for K seconds. During the stream, he received N Super Chats. Given the purchase time T_i and price P_i of each Super Chat. Can you determine for how long (in seconds) each Super Chat has displayed at the top of the chat window? Note: The ending of the stream has no effect on existing Super Chats. Pinned Super Chats will continue to count down normally until they expire. There will be no new Super Chats once the stream ends.

Figure 1: Irrelevant to the task.

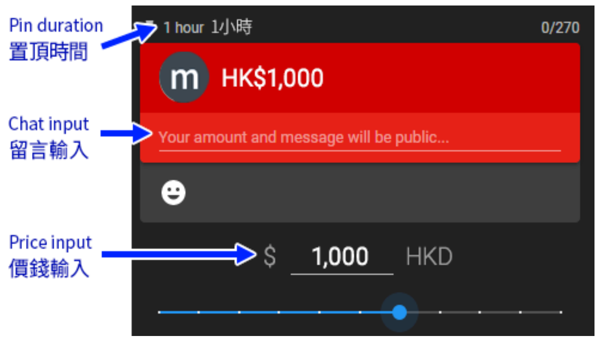


Figure 2: Super Chat Purchase user interface

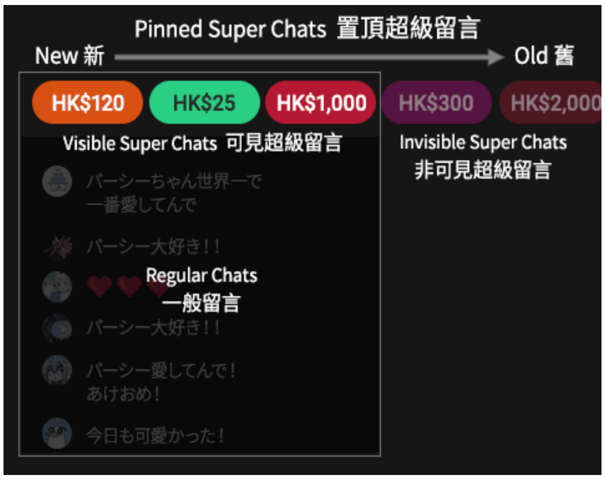

Pinned Super Chats 置頂超級留言
New 新
Old 舊
HK$120
HK$25
HK$1,000
HK$300
HK$2,000
Visible Super Chats 可見超級留言
Invisible Super Chats 非可見超級留言
Regular Chats 一般留言
パーシーちゃん世界一で一番愛してんで
パーシー大好き！！
パーシー大好き！！
パーシー愛してんで！あけおめ！
今日も可愛かった！

Figure 3: Chat window showing pinned Super Chats

INPUT

The first line contains an integer N -- the number of Super Chats.

The second line contains an integer K -- the total length of the stream (in seconds).

The i^{th} of the next N lines contains two integers T_i and P_i that describe the i^{th} Super Chat -- the time in seconds since the beginning of the stream when it is purchased, and its price respectively. ($5 \le P_i \le 2500$)

The input is given in increasing T_i. No two Super Chats are purchased at the same time. ($0 \le T_1 < T_2 < \cdots < T_i < T_{i+1} < \cdots < T_N < K$)

OUTPUT

Output N lines. On the i^{th} line output a single integer -- the number of seconds that the i^{th} Super Chat is visible at the top of the chat window.

SAMPLE TESTS

1

Input
```
7
300
0 25
35 25
70 25
110 25
140 25
150 25
210 25
```

Output
```
110
105
80
100
120
120
120
```

This sample satisfies the constraints of Subtask 1.
Time period when each Super Chat is visible:
Super Chat 1: 0 - 110
Super Chat 2: 35 - 140
Super Chat 3: 70 - 150
Super Chat 4: 110 - 210
Super Chat 5: 140 - 260
Super Chat 6: 150 - 270
Super Chat 7: 210 - 330 (The stream ended before the Super Chat expires)

2

```
4
4000
0 500
1 250
2 100
3 100
```

```
3001
1800
600
600
```

This sample satisfies the constraints of Subtask 2.

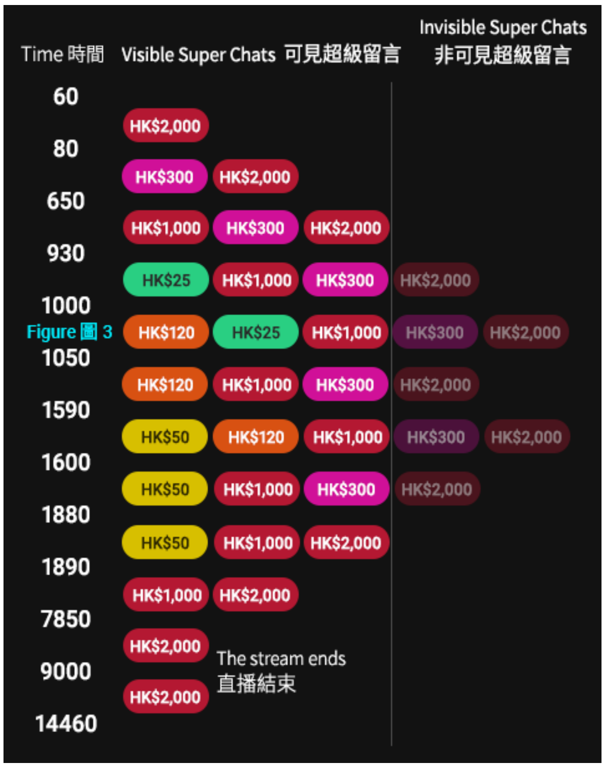

Time 時間
Visible Super Chats 可見超級留言
Invisible Super Chats 非可見超級留言
60
80
650
930
1000
Figure 圖 3
1050
1590
1600
1880
1890
7850
9000
14460
HK$2,000
HK$300 HK$2,000
HK$1,000 HK$300 HK$2,000
HK$25 HK$1,000 HK$300 HK$2,000
HK$120 HK$25 HK$1,000 HK$300 HK$2,000
HK$120 HK$1,000 HK$300 HK$2,000
HK$50 HK$120 HK$1,000 HK$300 HK$2,000
HK$50 HK$1,000 HK$300 HK$2,000
HK$50 HK$1,000 HK$2,000
HK$1,000 HK$2,000
HK$2,000
HK$2,000
The stream ends
直播結束

Figure 4: Explanation of Sample Test 3

3

7
9000
60 2000
80 300
650 1000
820 5
930 25
1000 120
1590 50
	13450
	1740
	7200
	0
	120
	600
	300

Please refer to Figure 4.

SUBTASKS

For all cases:
$1 \le N \le 200000$
$N \le K \le 10^9$
$5 \le P_i \le 2500$

	Points	Constraints
1	16	$P_i = 25$, i.e. the pin duration of each and every Super Chat is 2 minutes. $1 \le N \le 200000$ $N \le K \le 500000$
2	15	$N = 4$ $4 \le K \le 20000$
3	18	$1 \le N \le 1000$ $N \le K \le 20000$
4	27	$1 \le N \le 200000$ $N \le K \le 500000$
5	24	No additional constraints

S213 - CHINESE CHECKERS

Time Limit: 1.000 s / Memory Limit: 256 MB

Cherry loves playing Chinese Checkers (波子棋 / 中國跳棋). Every month she would hold a Chinese Checkers party and invite all her friends to come. Unfortunately these parties had to be cancelled in 2020, Cherry had to play alone with the *Capture* variant, placing all 60 marbles in the centre of the gameboard and capture as many marbles as possible using jumping moves.

Figure 1: Chinese Checkers

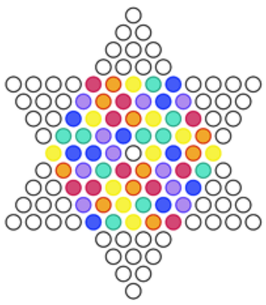


Figure 2: *Capture* variant

Attending classes from home makes Cherry feel bored, so she created a *Capture* variant that she can play on her own.

The game is to be played on a $(R+2) \times (C+2)$ rectangular checkerboard $(R, C \geq 2)$, where $(0,0)$ is top left and $(R+1, C+1)$ is bottom right. $R \times C$ marbles are initially placed in cells $(1,1)$ to (R,C), i.e. rows 0 and $(R+1)$ and columns 0 and $(C+1)$ are empty, and all other cells are placed with marbles.

In each move, you pick a marble and *jump* over a neighbouring marble to land on an empty cell, and the marble being jumped over will be removed from the board. Two marbles are neighbours if their corresponding cells share a common edge or a common corner. In mathematical sense, if you pick the marble in cell (r,c), and cell $(r+\Delta r, c+\Delta c)$ also contains a marble $(|\Delta r| \leq 1$ and $|\Delta c| \leq 1)$, your marble can be jumped to empty cell $(r+2\Delta r, c+2\Delta c)$, and both cells (r,c) and $(r+\Delta r, c+\Delta c)$ will become empty. Of course marbles cannot jump out of the board.

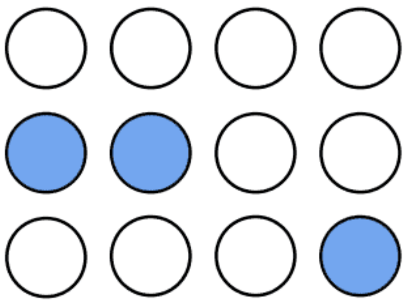


Figure 3; Animation of two jumps.

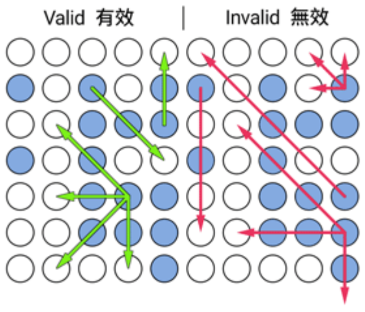


Figure 4: Some examples of valid and invalid jumps.

Cherry is very satisfied with this new game, so she sends this game to Bob to test it out. One hour later Bob replies with an astonishing fact: it is possible to remove all marbles except one (no more marbles for it to jump over), regardless of the size of the board! Bob even challenges Cherry back to prove his claim.

Since you have joined a lot of Chinese Checkers parties before, Cherry reaches to you for help. Can you come up with a strategy to remove as many marbles as possible? A simulator is provided at the end of this page.

INPUT

The first and only line contains two integers, R and C.

OUTPUT

On the first line output N, the number of moves you are going to make. $(1 \leq N \leq R \times C - 1)$

Then output N lines, the i^{th} of these N lines contains four integers $r1_i$, $c1_i$, $r2_i$, and $c2_i$, representing the coordinates of the marble $(r1_i, c1_i)$ to jump and the marble $(r2_i, c2_i)$ to be jumped over in the i^{th} jump.

If there are multiple sets of jumps, output any of them.

SCORING

For each test case, when your strategy contains N moves, that leaves $M = (R \times C - N)$ marbles on the board.

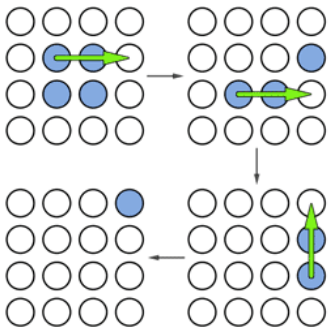
The score you get for the test case is $40 \times \frac{1}{\sqrt{M}} + 10^{1-\frac{M-1}{\min(R,C)}} + 50^{1-\frac{M-1}{R \times C}}$ rounded down to the nearest integer.

Your score for each subtask is the minimum score of all test cases in the subtask, multiplied by the point value of the subtask.

SAMPLE TESTS

	Input	Output

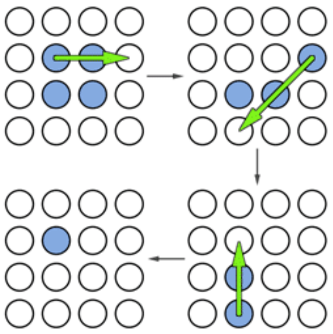
1	2 2	3
		1 1 1 2
		2 1 2 2
		2 3 1 3



	Input	Output

2	2 2	3
		1 1 1 2
		1 3 2 2
		3 1 2 1

Another set of moves for 2x2 marbles.



SUBTASKS

For all cases: $2 \le R, C \le 100$

	Points	Constraints
1	2	$R = 2, C = 2$
2	3	$R = 2, C = 3$
3	4	$R = 3, C = 3$
4	5	$R = 4, C = 3$
5	6	$R = 4, C = 4$
6	14	$2 \le R, C \le 5$
7	15	$R = 2$
8	19	$R = 3$
9	32	No additional constraints