SYSU Novice Programming Contest 2018

Registration site

The contest registration will be available:

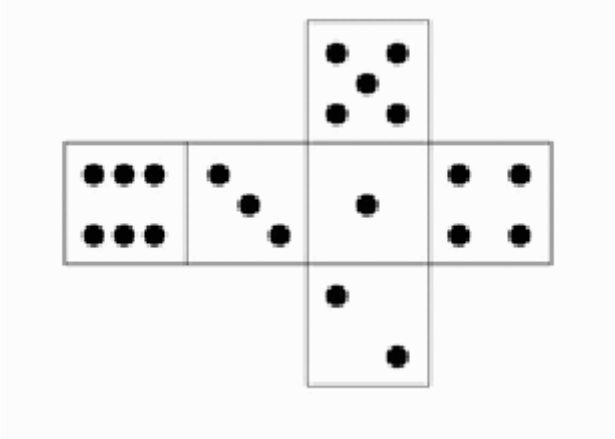
2018-12-05 16:00:00 -- 2018-12-14 00:00:00

A.Pug's Dice Tower

Time Limit : 1000 ms
Memory Limit : 262144 KB

Description

Pug noticed an interesting phenomenon of dice: the sum of dots on any two opposite sides equals 7.



[die.png]

Pug puts a die on the table and then (if he wants) he adds more dice, each time stacking a new die on the top of the tower. The dice in the tower are aligned by their edges so that they form a perfect rectangular parallelepiped. The parallelepiped's height equals the number of dice in the tower and two other dimensions equal 1 (if we accept that a die's side is equal to 1).

Pug's aim is to build a tower of minimum height and the sum of points on all its outer surfaces should equal the given number n (outer surface: the side surface, the top and bottom faces).

Write a program that would determine the minimum number of dice in the required tower by the given number n. Pug can construct any towers whose height equals 1 or more.

Input

The only input line contains integer n ($1 \leqslant n \leqslant 10^6$).

Output

Print the only integer - the number of dice in the required tower. If no such tower exists, print -1.

Examples

input
50
output
3

input
7
output
-1

input
32
output
2

B.Disillusion's Flappy Bird

Time Limit: 1000 ms
Memory Limit: 131072 KB

Problem Description

Disillusion is found of Flappy Bird. The bird starts at point (0,0), aiming at positions whose x-coordinate equal to X. (Which is the line x=X.)

Each second Disillusion can make a choice, tap and fly to (x+1,y+1) from (x,y), or let it fall to (x+1,y-1).

There's no ground but n barriers. The i-th of them lies on the line x=x[i] and occupies the space y<=a[i] and y>=b[i]. They do not have thickness.

If the bird arrives at point (x,y) (x=x[i],y>=b[i] or y<=a[i]) then Disillusion is failed.

Disillusion is lazy, so please tell him the minimum times of tapping required to get the target. If Disillusion is impossible to get there, ouput "GG" (without quote).

Input

Two integer on the first line n(0<=n<=500000),X(1<=X<=10^9).
For the next n lines,each have three integers,x[i],a[i],b[i](0<x[i]<X, -10^9<=a[i]<b[i]<=10^9).
It is guaranteed that x[i]<x[i+1].

Output

The minimum times of tapping required to get the target.
If impossible, output "GG" (without quote).

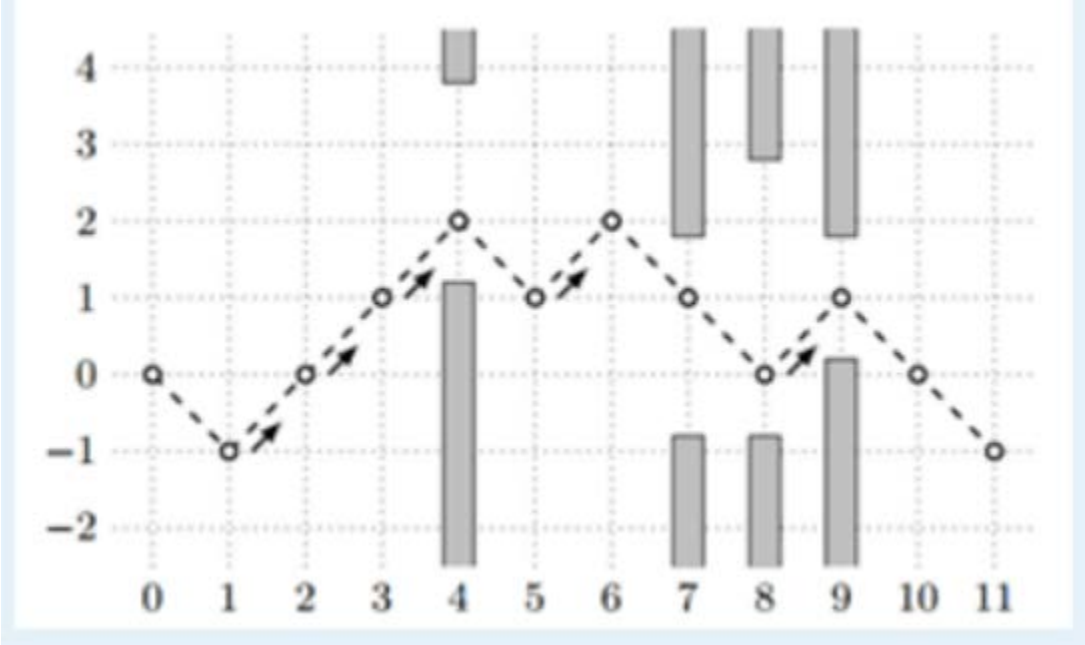
Sample Input

```
4 11
4 1 4
7 -1 2
8 -1 3
9 0 2
```

Sample Output

```
5
```

Hint



[Hint.png]

C.Di4CoveRy's Query

Time Limit : 10000 ms
Memory Limit : 262144 KB

Description
Di4CoveRy:"Can you solve this problem? Given a array of length n. I'll query the sum of some intervals. Also I'll modify some elements."

Hemp:"How can I do that?"

Di4CoveRy:"You can split the array into sqrt(n) parts. Each part has sqrt(n) elements. Get the sum of each part. Then if I try to modify one element, just calculate the sum of the part it belongs to again. If I query a interval, I can check each part whether it's in the interval. If all the elements of it are in the interval, then I can just add the sum I have calculated previously to the answer. If they are partly in the interval, I should go into the part and use a loop to solve this. So I got a O(sqrt(n)) algorithm for each operation."

Hemp:"Sounds that it works. /*I understood nothing.*/"

Please help Hemp with the problem.

Input
The first line contains two integers n,m.
"n" is the length of the array. "m" is the number of operations.
The second line contains n integers, which is the value of the elements in array.
Then follows m lines. There are three integers a,b,c.
"a" equals to 1 means Di4CoveRy wants to query the sum of the inteval [b,c].
otherwise(a=2) means Di4CoveRy changes the b-th element to c.

1<=n<=1e6 1<=m<=1e5 0<=a[i]<=1e3 (for any i any time)

Output
For each operation 1, print a line with the answer.

Sample input
3 3

1 2 3
1 1 3
2 2 4
1 1 3

Sample output
6
8

Hint
You may need a fast IO.
Here's an example for c(c++).

```
inline void read(int &x)
{
    x=0;
    static char ch=getchar();
    for (;(ch<'0' || '9'<ch) && ch!=EOF;ch=getchar());
    for (;'0'<=ch && ch<='9' && ch!=EOF;ch=getchar())
x=(x<<3)+(x<<1)+(ch^48);
}
```

D.Ryan's Pyramid

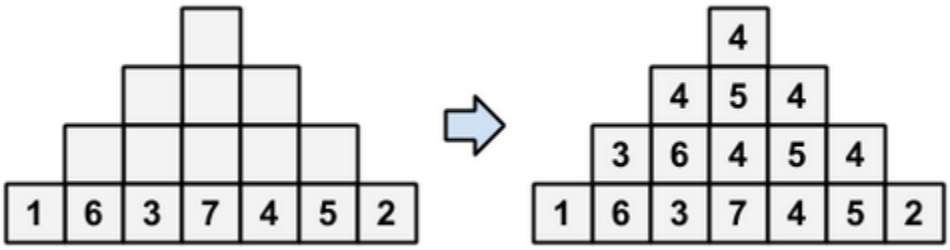
Time Limit : 2000 ms
Memory Limit : 262144 KB

Description

Ryan is the god of sun. He wants to build a pyramid. He got 2n-1 stones with unique id (a permutation of 1..2n-1) in a line and use his magic.

The magic created a new floor of x stones from the original top x+2 stones.

In other words every three adjacent stones create a new one with their ids' median as its id.



[pyramid.png]

Ryan repeat his magic until there's only one stone. Then he question you about the id of that stone.

Input

The standard input is as follows (all are integers)

N ($2 \leqslant N \leqslant 10^5$)

a_1 a_2 … a_{2N-1}

Output

one integer

Sample Input 1

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

Sample Output 1

```
4
```

Sample Input 2

2
1 2 3
Sample Output 2
2

E.PureWhite's DAG

Time Limit : 1000ms
Memory Limit : 131072KB

PureWhite is a person without sense of direction. He is lost in a DAG[*1] again.

[1]A DAG is a graph with n vertexes, m directed edges[*2] without a round. Which means for any vertex, if you start at it and walk alone the directed edges, you will never be able to go back to your starting point.

[2] A directed edge is a one-direction-passable edge. For example, if there's only a directed edge from u to v, you can move from u to v but not v to u.

The edges may have different length. PureWhite wonders if he starts at point s and end at point t, what is the minimum W=(total length of the edges he past)/(the number of the edges he past) he could get.

PureWhite has no idea about where he comes from and where he is going to. So he would ask multiple questions. These questions do not effect each other when calculating something about the edges he past.

Input
The first line contains two integers n,m.
Then follows m lines with three integers u,v,w, which means there's an edge from u to v with the length of w.
Next line will be a integer Q indicates the number of questions.
Then follow's Q lines. Each line has two integers s and t.

1<=n<=50 1<=m<=1000 1<=w<=100000 1<=Q<=100000

Output
Print W to the third decimal (Each query occupies a line.).
Or print "OMG!" if s cannot achieve t (without quote).

Sample Input
3 3
1 3 5

2 1 6
2 3 6
2
1 3
2 3

Sample Output
5.000
5.500