NOI 2016 TASKS

Tasks
Task 1: LunchBox
Task 2: Fabric
Task 3: Rock Climbing
Task 4: Panda Ski
Task 5: UnluckyFloors

Notes:

1. This document consists of 12 pages excluding this page.
2. Each task is worth 100 marks.
3. For each task, your program will be tested on a few sets of input instances. We call each set a subtask. Each subtask is worth a few marks.
4. The subtasks vary in size and complexity, leading to different levels of difficulty. If you find solving the task completely difficult, you may want to focus on the easier subtasks.
5. The maximum execution time on each input instance in Tasks 1, 2, 3, 4 & 5 are 0.5, 1.0, 1.0, 1.0 & 2.5 seconds, respectively, and the memory size is limited to 128MB for all tasks.

HAPPY PROGRAMMING!

Task 1: `LunchBox`

You are the manager of a restaurant. You prepare N lunch boxes and hope to distribute them to some schools. Suppose there are m schools and assume the ith school asks for k_i lunch boxes.

You aim to distribute the lunch boxes to as many schools as possible. Moreover, you have a rule. For the ith school, you give either zero or k_i lunch boxes. Can you make a program that help you to find the maximum number of schools that can receive lunch boxes?

Input

Your program must read from standard input. The first line contains 2 integers, N and m. Then, it follows by m lines. The ith line contains an integer k_i.

Output

Your program must output one line with a single integer to the standard output, which is the maximum number of schools.

Sample Testcase

Consider the following input:

```
10 4
3
9
4
2
```

In this example, the answer is 3 since $3+4+2 \leq 10$ and $3+9+4+2 > 10$. For this example, the output is:

```
3
```

Subtasks

The maximum execution time on each instance is 0.5s. Your program will be tested on sets of input instances that satisfies the following restrictions:

Subtask	Marks	Restrictions
1	20	Each instance satisfies $m = 1$, $0 < N \leq 60\,000$ and $0 < k_i \leq 30\,000$
2	30	Each instance satisfies $0 < m \leq 1\,000$, $0 < N \leq 60\,000$ and $0 < k_i \leq 1\,000$
3	50	Each instance satisfies $0 < N, m \leq 60\,000$ and $0 < k_i \leq 30\,000$

Task 2: `Fabric`

Kraw the Krow has a beautiful piece of fabric. The patterns are so intricate that every part of the fabric is different. However, after the Great Fire of 2017, the fabric now has a lot of unsightly holes. (The Great Fire was started, of course, by none other than Squeaky the Rat.)

Kraw wants to forget about the Great Fire, because he doesn't like heat very much. He would like to cut out a rectangle of fabric and throw the rest away. The new piece of fabric must have an area of at least K and cannot contain any holes.

Due to the gauge-antisymmetric properties of Kraw's fabric (or something – Kraw can't remember what the salesman said), Kraw can only cut the fabric along regular gridlines. Kraw wonders how many ways there are to cut a rectangle with an area of at least K out of the fabric such that it contains no holes.

Input format

Your program should read the input from standard input. The input consists of:

- one line with three integers N and M ($1 \leq N, M \leq 2\,000$), the height and width of the fabric, and K ($1 \leq K \leq MN$), the minimum area of the rectangle in terms of the number of grid segments it must contain;
- N lines each with M integers $s_{0y}, s_{1y}, \ldots, s_{(M-1)y}$. s_{xy} is 1 if there is a hole on the segment with coordinates (x, y), and 0 if there is no hole.

An example is as follows:

```
2 4 3
1 0 0 0
0 0 0 1
```

Output format

Output one line with a single integer: the number of ways to cut a rectangle with an area of at least K out of the fabric such that it contains no holes.

For the input above, the output is:

```
3
```

Explanation

3 rectangles with an area of at least 3 can possibly be cut from the fabric. Taking the top left segment as (0, 0), there are:

- 2 rectangles of area 3 – {(1, 0), (2, 0), (3, 0)}, {(0, 1), (1, 1), (2, 1)}
- 1 rectangles of area 4 – {(1, 0), (2, 0), (1, 1), (2, 1)}

Subtasks

The maximum execution time on each instance is 1.0s. Your program will be tested on sets of input instances as follows:

Subtask	Marks	Restrictions
1	7	Each instance satisfies $0 < N, M \leq 2000$, $K = 1$ and $s_{xy} = 0$ for all (x, y);
2	9	Each instance satisfies $0 < N, M \leq 2000$, $K = 1$ and $s_{xy} = 1$ for only one (x, y);
3	12	Each instance satisfies $0 < N, M \leq 50$;
4	14	Each instance satisfies $0 < N, M \leq 500$;
5	23	Each instance satisfies $0 < N, M \leq 2000$ and $K = 1$;
6	35	Each instance satisfies $0 < N, M \leq 2000$.

Task 3: `Rock Climbing`

On one fine day, Mr. Panda and Rar the Cat decide to go rock climbing. The rock climbing wall has N rocks. The i-th rock is located at height Y_i from the bottom of the wall and X_i units right from the centre of the wall. If X_i is negative then it is to the left of the centre. The positions of all rocks are different.

To test Mr. Panda's rock climbing skills, Rar the Cat decided to issue a challenge to him. The challenge is as follows:

1. Out of the N rocks, Rar the Cat will pick a set of K rocks, this set is called R.
2. In order to win the challenge, Mr. Panda first has to pick one pair of rocks (A, B) from set R. Mr. Panda is free to choose any pair of rocks as long as $A \neq B$ and both rocks are in set R.
3. Mr. Panda will start from the first rock (A) and try to make his way to the second rock (B). He can pass through other rocks on the way from A to B, regardless of whether the rock is in set R.
4. However, each rock is associated with a slippery rate S_i. When a rock has a high slippery rate, it is more difficult to stretch to a far away rock without slipping off. Furthermore, Rar the Cat only allows him to climb upwards. More precisely, to move from the ith rock to the jth rock, we require $\max(|X_i - X_j|, |Y_i - Y_j|) \leq \max(S_i, S_j)$ and $Y_i < Y_j$
5. Mr. Panda will win the challenge if he manages to pick one pair of rocks (A, B) such that he can get from rock A to rock B. If he fails to do that, Mr. Panda would have lost the challenge.

Refer to the sample input and output for more details.

Of course, Mr. Panda knows that there are many pairs of rocks such that the challenge cannot be completed. He wants to find the minimum K such that no matter what set of rocks Rar the Cat chooses, he can always complete the challenge. He needs your help to find this value.

Input format

Your program must read from standard input. The first line of input contains one integer N. The next N lines contain 3 integers each. The $(i + 1)$-th line represents X_i, Y_i, S_i for the i-th rock. A possible input would be:

```
5
0 3 2
-1 5 1
4 4 3
-1 1 2
2 2 1
```

Output format

You program must output one line with a single integer to the standard output, which is the minimum number of rocks so that Mr. Panda can always complete the challenge. If Mr. Panda can never complete the challenge, output -1. For the above example, the answer is:

```
3
```

Explanation

When $K = 2$, there exists some subset of rocks so that Mr. Panda cannot complete the challenge. For example, if Rar the Cat chooses the rocks on $(-1, 1)$ and $(2, 2)$ to form the set R, Mr. Panda cannot complete the challenge.
Mr. Panda cannot move from the rock on $(2, 2)$ to the rock on $(-1, 1)$ since he can only climb upward. Mr. Panda cannot move directly from the rock on $(-1, 1)$ to the rock on $(2, 2)$ since $\max(|-1-2|, |1-2|) = 3 > \max(2, 1) = 2$.

Another choice is to move indirectly from $(-1, 1)$ to $(2, 2)$. Mr. Panda can climb from the rock on $(-1, 1)$ to the rock on $(0, 3)$ since $\max(|-1-0|, |1-3|) = 2 \leq \max(2, 1) = 2$. However, he cannot move from the rock on $(0, 3)$ to the rock on $(2, 2)$ since he must climb upwards.

Moreover, it can be verified that for every set of 3 rocks, there is always a way to pick a pair of rocks so that the Mr. Panda can complete the challenge.
For example, if Rar the Cat picks the rocks on $(-1, 5), (4, 4), (-1, 1)$ to form set R, Mr. Panda can complete the challenge by picking the rocks on $(-1, 5)$ and $(-1, 1)$.
To complete the challenge, he climbs from the rock $(-1, 1)$ to the rock on $(0, 3)$ then to the rock on $(-1, 5)$. The intermediate rocks do not need to be from R.

Sample Input 2

This testcase is only valid for subtasks 1,2 and 5

```
3
0 1 2000000
-1 2 2000000
1 2 2000000
```

Sample Output 2

```
3
```

Explanation

If Rar the Cat selects the 2 rocks on height 2 (i.e. the 2nd and the 3rd rocks), Mr. Panda cannot complete the challenge since he must always climb upwards. So Mr. Panda can only guarantee that he can complete the challenge if Rar the Cat chooses all the rocks.

Sample Input 3

This testcase is only valid for subtasks 2,3,4 and 5

```
2
0 1 1
0 5 1
```

Sample Output 3

```
-1
```

Explanation

Even if Rar the cat selects all the rocks Mr. Panda since there is no way to get from one rock to another. Thus, Mr. Panda can never complete the challenge.

Subtasks

The maximum execution time on each instance is 1.0s. Your program will be tested on sets of input instances as follows:

Subtask	Marks	N	Others
1	15	$1 \leq N \leq 20$	$S_i = 2 \times 10^6$ for all i
2	29	$1 \leq N \leq 20$	No other restriction
3	17	$1 \leq N \leq 500$	$X_i = 0, S_i = S_j$ for all i, j
4	19	$1 \leq N \leq 500$	$S_i = 1, Y_i$ is not a multiple of 3 for all i
5	20	$1 \leq N \leq 500$	No other restriction

For all test cases, $-10^6 \leq X_i \leq 10^6$, $1 \leq Y \leq 10^6$, $1 \leq S_i \leq 2 \times 10^6$

Task 4: `Panda Ski`

The Winter Olympics is coming and Mr. Panda has been training very hard to take part in the skiing event. This event takes place on the mountain Mt. Rar which is of height H. Everyone can ski down from the peak to the base using the centroid path. To increase the difficulties, N gates, each associated with a score, are placed at various heights and either to the left or to the right of the centroid path. The objective is to ski down from the peak to the base and achieve score by passing through some subset of gates.

The i-th gate is located at height Y_i and X_i units to the right of the centroid path. If X_i is negative then it is to the left of the centroid path. Passing through the i-th gate gives S_i points and you can pass through the same gate multiple times but you only get points for the first time you pass through a gate. No gate is at the same point.

Mr. Panda wants to maximize his score. Moreover, Mr. Panda understands he is not a good skier and he will fail to visit some gates. To avoid embarrassing himself, Mr. Panda analyses the gates and gives each gate an easiness score E_i (high score is easier) based on the angle of the slope, the amount of snow, etc.

Precisely, Mr. Panda calculated that he can move from the i-th gate to the j-th gate if $\max(|X_j - X_i|, Y_i - Y_j) \leq E_i$ and $Y_i \geq Y_j$. Also, it is possible to get from the peak to any gate and from any gate to the base of the mountain.

Mr. Panda is overwhelmed by the number of possible paths moving down the mountain and he needs your help to find the path that will give him the maximum score.

Input format

Your program must read from standard input. The first line of input contains two positive integers N and H. The next N lines contain 4 integers each. The $(i + 1)$-th line represents X_i, Y_i, S_i, E_i. A possible input would be:

```
5 5
0 5 5 1
3 4 4 3
-2 3 3 2
1 1 4 4
-1 2 3 1
```

Output format

You program must output one line with a single integer to the standard output, which is the maximum score Mr. Panda can attain. For the above example, the answer is:

8

Explanation

There are only 3 possible paths Mr. Panda can take.

1. Top $\rightarrow (0,5) \rightarrow$ Bottom, Score: 5
2. Top $\rightarrow (3,4) \rightarrow (1,1) \rightarrow$ Bottom, Score: $4+4=8$
3. Top $\rightarrow (-2,3) \rightarrow (-1,2) \rightarrow$ Bottom, Score: $3+3=6$

So the best score is 8.

Subtasks

The maximum execution time on each instance is 1.0s. Your program will be tested on sets of input instances as follows:

Subtask	Marks	N	Others
1	7	$1 \leq N \leq 300$	$E_i = 200000$ for all i
2	8	$1 \leq N \leq 300$	$X_i = 0$ for all i, $E_i = E_j$ for all i, j
3	11	$1 \leq N \leq 300$	$Y_i \neq Y_j$ for all $i \neq j$
4	13	$1 \leq N \leq 2000$	$Y_i \neq Y_j$ for all $i \neq j$
5	15	$1 \leq N \leq 50000$	$Y_i \neq Y_j$ for all $i \neq j$, $E_i = E_j$ for all i, j
6	13	$1 \leq N \leq 50000$	$E_i = E_j$ for all i, j
7	16	$1 \leq N \leq 50000$	$Y_i \neq Y_j$ for all $i \neq j$
8	17	$1 \leq N \leq 200000$	$Y_i \neq Y_j$ for all $i \neq j$

For all test cases, $-50000 \leq X_i \leq 50000$, $1 \leq Y_i \leq H \leq 200000$, $1 \leq S_i \leq 10^6$, $1 \leq E_i \leq 200000$

Task 5: `UnluckyFloors`

When Rar the Cat went to Taiwan for IOI 2014, he was accomodated in a hotel. During his stay, he realised that certain floors are 'missing' from the hotel building. Namely, he observed that numbers containing 4 and 13 as substrings are omitted from the floor numberings. This is because 4 and 13 are considered unlucky numbers and are purposely left out in the numbering. For simplicity, we will refer to this numbering scheme as the lucky numbering scheme, as it omits the unlucky numbers. The table below shows the first 20 floors in a lucky numbering scheme as well as the conventional numbering scheme.

Conventional	**Lucky**
1	1
2	2
3	3
4	5
5	6
6	7
7	8
8	9
9	10
10	11
11	12
12	15
13	16
14	17
15	18
16	19
17	20
18	21
19	22
20	23

However, Rar the Cat feels that such a numbering scheme is not legitimate and wants to be able to convert floors between the lucky and conventional numbering scheme. For example, floor 6 in the lucky numbering scheme will be floor 5 in the conventional numbering scheme and floor 15 will actually be floor 12. Hence, given a floor number in the lucky numbering scheme, Rar the Cat wants you to compute which floor it will be in the conventional numbering scheme and vice-versa.

Input

Your program must read from standard input.

The input will start with a single integer, N, in a single line. N denotes how many floor numbers Rar the Cat wants you to convert for him.

N lines will then follow with 2 integers each, the i^{th} line will contain T_i and X_i.

If T_i is 1, you are to convert X_i from the lucky numbering scheme to the conventional numbering scheme and print the result in a single line. However, if X_i is not a valid number in the lucky numbering scheme, print -1 as the answer instead.

If T_i is 2, you are to convert X_i from the conventional numbering scheme to the lucky numbering scheme and print the result on a single line.

Output

Your program must output to standard output only.
Output a total of N lines with 1 integer each. For each i, output the answer to T_i and X_i.
It is guaranteed that the answer will fit in a 64-bit signed integer. Refer to Sample Testcase 4 and 5 for more information.

Subtasks

The maximum execution time on each instance is 2.5s. Your program will be tested on sets of input instances that satisfies the following restrictions:

Subtask	Marks	N	T_i	X_i
1	5	$0 < N \leq 50$	$T_i = 1$ or 2	$0 < X_i \leq 25$
2	12	$0 < N \leq 50$	$T_i = 1$ or 2	$0 < X_i \leq 100\,000$
3	18	$0 < N \leq 100\,000$	$T_i = 1$ or 2	$0 < X_i \leq 100\,000$
4	11	$0 < N \leq 100\,000$	$T_i = 1$	$X_i = 10^K - 1$ where $1 \leq K \leq 16$
5	37	$0 < N \leq 100\,000$	$T_i = 1$	$0 < X_i \leq 10^{16}$
6	17	$0 < N \leq 100\,000$	$T_i = 1$ or 2	$0 < X_i \leq 10^{16}$

Sample Testcase 1

This testcase is only valid for subtasks 1, 2, 3, 5 and 6.

Input	Output
8 1 1 1 4 1 15 1 25 2 1 2 4 2 15 2 25	1 -1 12 21 1 5 18 29

Sample Testcase 2

This testcase is only valid for subtasks 2, 3 and 6 only.

Input	Output
10 1 1 2 4 1 15 2 15 1 26 1 131 2 131 2 1337 1 100000 2 100000	1 5 12 18 22 -1 178 1995 56160 190508

Sample Testcase 3

This testcase is only valid for subtasks 4, 5 and 6 only.

Input	Output
2 1 9 1 999999999999	8 245967827040

Sample Testcase 4

This testcase is only valid for subtasks 5 and 6 only.

Input	Output
5 1 987328938823 1 75732858587173 1 4444444444444444 1 1313131313131313 1 10000000000000000	241928778399 13999321852875 -1 -1 1534593233484559

Sample Testcase 5

This testcase is only valid for subtask 6 only.

Input	Output
5 2 987328938823 2 75732858587173 2 4444444444444444 2 1313131313131313 2 10000000000000000	5110985302888 500859079673722 30071998020860537 8755153350232701 7673211628595292 8