Problem 1: Mirror Field

Farmer John has left some old mirrors sitting outside his house, and his cows, feeling mischievous as always, have stolen them!

The cows have set up the mirrors in a rectangular field measuring N by M squares (1 <= N, M <= 1,000). In each square, they have placed a double sided mirror between two of its opposite corners. These two possible configurations are represented by the '/' character (a mirror connecting the lower-left corner to the upper-right corner) and the '\' character (a mirror connecting the upper-left corner to the lower-right corner).

One evening, Bessie the cow brings a laser pointer out to the mirror field. Standing outside the field, she shines the beam of light either horizontally or vertically along either a row or column of the field, causing it to bounce of some number of mirrors. Since the mirrors are all diagonally oriented, a horizontal beam of light that reflects off a mirror will end up traveling vertically, and vice versa. Bessie wonders what is the maximum number of mirrors on which her beam of light can be reflected at the same time. Given the layout of the mirror field, please help Bessie compute this number.

PROBLEM NAME: mirror

INPUT FORMAT:

* Line 1: The integers N and M, separated by a space.

* Lines 2..1+N: Each line will contain M '/' or '\' characters,
describing a row of the mirror field.

SAMPLE INPUT (file mirror.in):

```
3 3
/\\
\\\
/\/
```

OUTPUT FORMAT:

* Line 1: A single integer indicating the maximum number of times a

horizontal or vertical beam originating outside the mirror field could be reflected. Please output -1 if it could be reflected indefinitely.

SAMPLE OUTPUT (file mirror.out):

3

OUTPUT DETAILS:

Bessie can shine the beam downwards above the middle column of her field to have it reflected 3 times.

Problem 2: Auto-complete

Bessie the cow has a new cell phone and enjoys sending text messages, although she keeps making spelling errors since she has trouble typing on such a small screen with her large hooves. Farmer John has agreed to help her by writing an auto-completion app that takes a partial word and suggests how to complete it.

The auto-completion app has access to a dictionary of W words, each consisting of lowercase letters in the range a..z, where the total number of letters among all words is at most 1,000,000. The app is given as input a list of N partial words (1 <= N <= 1000), each containing at most 1000 lowercase letters. Along with each partial word i, an integer K_i is also provided, such that the app must find the (K_i)th word in alphabetical order that has partial word i as a prefix. That is, if one ordered all of the valid completions of the ith partial word, the app should output the completion that is (K_i)th in this sequence.

PROBLEM NAME: auto

INPUT FORMAT:

* Line 1: Two integers: W and N.

* Lines 2..W+1: Line i+1: The ith word in the dictionary.

* Lines W+2..W+N+1: Line W+i+1: A single integer K_i followed by a
  partial word.

SAMPLE INPUT (file auto.in):

```
10 3
dab
ba
ab
daa
aa
aaa
aab
abc
ac
```

dadba
4 a
2 da
4 da

OUTPUT FORMAT:

* Lines 1..N: Line i should contain the index within the dictionary
        (an integer in the range 1..W) of the (K_i)th completion (in
        alphabetical order) of the ith partial word, or -1 if there
        are less than K_i completions.

SAMPLE OUTPUT (file auto.out):

3
1
-1

OUTPUT DETAILS:

The completions of a are {aa,aaa,aab,ab,abc,ac}. The 4th is ab, which is listed on line 3 of the dictionary. The completions of da are {daa,dab,dadba}. The 2nd is dab, listed on line 1 of the dictionary. There is no 4th completion of da.

Problem 3:

Every morning, FJ wakes up and walks across the farm from his house to the barn. The farm is a collection of N fields (1 <= N <= 250) connected by M bidirectional pathways (1 <= M <= 25,000), each with an associated length. FJ's house is in field 1, and the barn is in field N. No pair of fields is joined by multiple redundant pathways, and it is possible to travel between any pair of fields in the farm by walking along an appropriate sequence of pathways. When traveling from one field to another, FJ always selects a route consisting of a sequence of pathways having minimum total length.

Farmer John's cows, up to no good as always, have decided to interfere with his morning routine. They plan to build a pile of hay bales on exactly one of the M pathways on the farm, doubling its length. The cows wish to select a pathway to block so that they maximize the increase in FJ's distance from the house to the barn. Please help the cows determine by how much they can lengthen FJ's route.

PROBLEM NAME: rblock

INPUT FORMAT:

* Line 1: Two space-separated integers, N and M.

* Lines 2..1+M: Line j+1 describes the jth bidirectional pathway in terms of three space-separated integers: A_j B_j L_j, where A_j and B_j are indices in the range 1..N indicating the fields joined by the pathway, and L_j is the length of the pathway (in the range 1...1,000,000).

SAMPLE INPUT (file rblock.in):

```
5 7
2 1 5
1 3 1
3 2 8
3 5 7
3 4 3
2 4 7
4 5 2
```

INPUT DETAILS:

There are 5 fields and 7 pathways. Currently, the shortest path from the house (field 1) to the barn (field 5) is 1-3-4-5 of total length 1+3+2=6.

OUTPUT FORMAT:

* Line 1: The maximum possible increase in the total length of FJ's
        shortest route made possible by doubling the length of a
        single pathway.

SAMPLE OUTPUT (file rblock.out):

2

OUTPUT DETAILS:

If the cows double the length of the pathway from field 3 to field 4 (increasing its length from 3 to 6), then FJ's shortest route is now 1-3-5, of total length 1+7=8, larger by two than the previous shortest route length.

Problem 4: Cow Decathlon

Farmer John's N cows (1 <= N <= 20), conveniently labeled 1...N as always, are preparing for a decathlon that has N different events (so perhaps it would be better called an N-athlon instead of a decathlon, which traditionally has exactly 10 events).

Cow i has a skill level of s_ij (1 <= s_ij <= 1000) when competing in event j. Each cow must compete in one and only one event, and each event must have some cow competing in it.

The total score for all cows is the sum of their skill levels for the events in which they are competing. However, the event judges can also give out bonus points if they are particularly impressed. There are B bonuses (1 <= B <= 20) that the judges can give out. Bonus i has three parts: if the cows obtain at least P_i points (1 <= P_i <= 40,000) for the first K_i events (including other bonuses involving just those events), they will get an additional A_i points (1 <= A_i <= 1000).

For example, let us consider N = 3 cows with the following skills:

```
     E V E N T
    | 1 | 2 | 3
  --+---+---+--
C 1 | 5 | 1 | 7
  --+---+---+--
O 2 | 2 | 2 | 4
  --+---+---+--
W 3 | 4 | 2 | 1
```

For example, cow 1 would earn the team 7 points if she participates in event 3.

Suppose the judges offer a bonus (B = 1), such that if the if the cows score at least 7 points in the first two events, they will get an additional 6 points. Here, the optimal assignment would be to assign cow 1 to event 1, cow 2 to event 3 and cow 3 to event 2. For the first two events, cow 1 will score 5 points and cow 3 will score 2 points giving them 7 points, which is enough to satisfy bonus 1. Therefore, the total points that they score will be 5+2+4+6=17.

Please help decide which events the cows should attempt to maximize their total score.

PROBLEM NAME: dec

INPUT FORMAT:

* Line 1: Two space-separated integers: N, B

* Lines 2..B+1: Line i+1 will contain the information for bonus i
  which is three space- separated integers: K_i, P_i, A_i.

* Lines B+2..B+N+1: Line B+1+j will contain the information on how cow
  j will perform at each of her events. This will be given in N
  space-separated integers: s_j1...s_jN.

SAMPLE INPUT (file dec.in):

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

OUTPUT FORMAT:

* Line 1: The maximum amount of points that the cows can receive,
  including bonuses.

SAMPLE OUTPUT (file dec.out):

```
17
```

OUTPUT DETAILS:

Cow 1 will perform event 1, cow 3 will perform event 2, and cow 2 will perform event 3.

Problem 5: Airplane Boarding

FJ's cows have decided to take a vacation, and have miraculously managed to find an airline willing to sell them tickets. When they arrive at the airport and start boarding their plane, they face an interesting problem, however.

The airplane has N seats, which we model as the points x=1 through x=N on the number line. All N cows (1 <= N <= 200,000) are standing in line waiting to get to their seats. Cow N is at position x=0, Cow N-1 is at position x=-1, and so on. Cow i has been assigned to Seat S_i, where S_1,...,S_N is a permutation of 1,...,N.

At each time step, each cow takes a step to the right if she can. When cow i reaches her seat S_i, she will stop to put her baggage in the overhead bin, which takes T_i seconds, and then sit down. For those T_i steps, the cow behind her (if there is one) is blocked from moving forward. If there is a line of cows behind her, the line is effectively blocked as well.

How long will it take for all the cows to sit down?

The sum of T_i for all cows will be less than 1,000,000,000.

PROBLEM NAME: boarding

INPUT FORMAT:

* Line 1: A single integer, N.

* Lines 2..N+1: Two space-separated integers, S_i and T_i.

SAMPLE INPUT (file boarding.in):

```
3
2 5
3 10
1 5
```

INPUT DETAILS:

Initially, the cows are situated like this:

```
cows -> 123
          123 <- seats
```

with cow 1 trying to get to seat 2, cow 2 trying to get to seat 3, and cow 3 trying to get to seat 1.

OUTPUT FORMAT:

* Line 1: A single line indicating the amount of time it takes to seat all cows.

SAMPLE OUTPUT (file boarding.out):

```
19
```

OUTPUT DETAILS:

After one step, they will all move 1 to the right and cow 3 will reach her seat:

```
 123
   123
```

Cow 3 takes 5 seconds to sit down, at which point she effectively disappears.

```
 12
   123
```

It takes 3 more seconds for cows 1 and 2 to reach their desired seats:

```
    12
   123
```

It takes 5 seconds for cow 1 to sit down and 10 seconds for cow 2 to sit down, so that's 10 seconds total.

In total this took 1 + 5 + 3 + 10 = 19 seconds.