IDI Open
Programming Contest
April 22nd, 2017

Problem Set

Jury and Problem Writers

Torbjørn Morland (Head Judge)
Karl Johan Heimark
Jean Niklas L'orange
Ruben Spaans
Edvard Kristoffer Karlsen

Tips

- Tear the problem set apart and share the problems among you.
- Problems are not ordered by difficulty.
- Try solving the easy problems first. Two problems in this set are tagged with "(Easy)" to help point you in the right direction.
- If your solution fails on a problem, you can print your program and debug it on paper while you let someone else work on a different problem on the computer.
- If you need help, contact the judges.

Rules

- Each team consists of one to three contestants.
- One computer is used per team.
- You may not cooperate with persons not on your team.
- You may print your programs on paper to debug them.
- What you may bring to the contest:
  - Any written material (Books, manuals, handwritten notes, printed notes, etc).
  - Pens, pencils, blank paper, stapler and other useful non-electronic office equipment.
  - NO material in electronic form (CDs, USB pen and so on).
  - NO electronic devices (PDAs and so on).
- The only electronic content you may consult during the content is that specified by the organiser (see the web-page). You may not copy source code from web pages, etc.
- Your programs should read from standard in and write to standard out. Writing to standard error will result in a failed submission. C programs should return 0 from `main()`.
- Your program may use at most 100MB of memory.
- Your programs may not:
  - access the network,
  - read or write files on the system,
  - talk to other processes,
  - fork,
  - or similar stuff.
  - If you try, your program will hang or crash. If it hangs, it will take a couple of minutes before others will be able to run their programs. So please make an effort to not crack/break what we have spent our spare time preparing for you.
- Show common sense and good sportsmanship.

Travelling Delivery Person
Problem ID: deliveryperson

Frederic Edward Xavier runs a very successful delivery company called FredEX. He's been looking at driving records lately, and is not happy with the costs of delivery.

He has made the following estimations:

- Driving from one intersection to the neighboring one costs B dollars
- Going straight through an intersection costs S dollars
- Turning right in an intersection costs R dollars
- Turning left in an intersection costs L dollars
- Delivering a package is free, but you still pay the cost for the turn/going straight in that intersection.

FredEX operates in a city where the streets make up an infinite grid of square blocks, with intersections numbered from $-\infty, -\infty$ to ∞, ∞. Lower numbers on the x axis are to the left, and lower numbers on the y axis are down. The truck starts at the intersection $0, 0$. All packages must be delivered in the specified order. The truck can not go in reverse, and cannot perform u turns. Help Frederic find out the cheapest way to deliver all N packages.

Input

The first line of the input is a line with five space separated integers, B, S, R, L, N.
Then follows N lines, each with two space separated integers, X_i and Y_i, the x and y coordinates of the intersection for delivering package number i.

Output

Output the minimum cost of delivering all packages.

Limits

- $1 \leq S, R, L, B \leq 100$
- $1 \leq N \leq 30\,000$
- $-5 \leq X_i, Y_i \leq 5$

Sample Input 1	Sample Output 1
1 1 1 10 2 2 2 1 2	14

Lava
Problem ID: lava

While walking in grand malls little Elsa likes to play a game she calls "Lava". This is a simple game using the different colors of the floor tiles as markers for where it is safe to walk and where it is not. The rules of the game say that you're allowed to walk on white tiles, but not on black tiles. The tiles are square.

Elsa typically plays this game together with her father. As he is an adult and (sometimes) can move over tiles that Elsa cannot, Elsa has decided that she is allowed to move in all directions while her father is only allowed to move along the 2 axes of the floor tiles, and they are only allowed to take steps up to a certain length. This means in particular that Elsa can move to any tile within the radius of her step length (in the usual Euclidean distance).

The goal of this game is to get from store A to store B as quickly as possible. Both players do each move at the same time and they are not allowed to do the next move until they are both finished with the first move. As both the father and Elsa hate to lose, the best outcome is to end up with them getting to the finish at the same time. The father would not like to participate in any games where Elsa will lose or a game where her father can be seen as trying to lose on purpose (Elsa always knows). A player who can't reach the goal obviously loses.

Input

The fist line of the input is a line consisting of 2 integers A and F indicating the maximum step length of Elsa and the maximum step length of the father respectively. The step lengths are given as multiples of a tile width.
Then follows a line with 2 integers L and W indicating the length and width of the map of their possible Lava game.
Then follow L lines each consisting of a string of length W, indicating the tiles of the map. Here 'S' denotes the start tile, 'G' denotes the goal tile, 'W' denotes a safe tile and 'B' denotes a lava tile.

Output

Output a line with the word "`SUCCESS`" if they both reach the goal at the same time, "`GO FOR IT`" if Elsa wins and "`NO CHANCE`" if the father would win. If neither Elsa nor her father is able to reach the goal, output "`NO WAY`".

Limits

- $1 \leq A \leq 1\,000$
- $1 \leq F \leq 1\,000$
- $1 \leq L \leq 1\,000$
- $1 \leq W \leq 1\,000$
- The number of 'W' tiles will be at most $1\,000$
- The map will only contain the characters 'S', 'G', 'W' and 'B' and there will only be one 'S' character and one 'G' character per map.

Sample Input 1	Sample Output 1
2 3 4 4 WWWW WSBB WWWW WBWG	GO FOR IT

Sample Input 2	Sample Output 2
1 1 1 2 GS	SUCCESS

Toys
Problem ID: toys

Little Lark has a great collection of of toys. Although she has plenty of toys she only likes to play with one at a time. She decides which toy to play with by placing all the toys in a circle around her, numbering them 0 to $T-1$. She then spins around in the clockwise direction and removes every Kth toy until one remains. This means that the first toy that she removes is numbered $K \bmod T$. If any toys are moved during this ritual Lark will start crying and then rearrange the toys in the circle in the original order.

Today Lark wants her father to join her while playing with the toys. Her father has of course a favorite toy among Lark's selection of toys and would of course like that specific toy to be chosen. At which position should he place his favorite toy to make sure that is the toy they end up playing with?

Input

The input is a single line consisting of 2 integers T and K indicating the number of toys Lark has and the skip length she takes while selecting the next toy to discard.

Output

Output a line with the a single integer, the position the father needs to place his favorite toy for it to be selected. Lark will start counting at position 0.

Limits

- $1 \leq T \leq 10\,000\,000$
- $1 \leq K \leq 1\,000\,000$
- $K \leq T$

Sample Input 1	Sample Output 1
`5 2`	`2`

Sample Input 2	Sample Output 2
`25 18`	`1`

Poetry Tower
Problem ID: poetrytower

Alice and Bob are young, aspiring poets. Alice likes variation in poems, so she only gets impressed by a poems that do not contain repeated words (and of course, the words used have to be real words).

Bob has found N six-sided blocks where each side has a letter or a wildcard symbol that can be used to represent any letter. He wants to use them to impress Alice with some really tall poetry. He has decided to put some blocks on top of each other to make a poem, and recite it to her. He thinks this is such a good idea that he has decided to do this for all possible poems Alice would be impressed by. Bob only knows M different real words. Help him figure out how many different poems he could impress Alice with using these blocks.

Note that even though the poems “to be or not” and “tob eor not” can be built in exactly the same way, they count as two poems because Bob only cares about how many poems he can recite, not how many different towers he can build.

Input

The first line of the input is a single line with two space separated integers N and M.
The next M lines constitute the words that Bob knows. Each line has a single word w.
The next N lines each contain a string of six characters representing a die.

Output

Output the number of different impressive poems Bob can build and recite with the provided blocks.

Limits

- $1 \leq N \leq 20$
- $1 \leq M \leq 8$
- Each word that Bob knows is between 1 and 20 characters long
- All words that Bob knows are unique
- Each word that Bob knows consists only of the letters a-z
- The sides of each die are only the letters `a-z` or $\star$ (representing a wildcard)

Sample Input 1	Sample Output 1
6 4 abc idi open def idiope nabcde *abcde bcdefg cccccc ******	8

Johnny Applesack
Problem ID: applesack

Johnny Applesack has grown his seeds into trees and harvested all the apples. He has a huge pile of N apples next to him, a sack that can hold K apples at a time, and an infinitely long, straight road in front of him. At each kilometer mark of the road there is a toll booth. If he passes the toll booth in the direction that brings him further away from where he started, he has to pay one apple. Otherwise he will not be allowed to pass. If he passes the other way, he does not have to pay. Johnny is allowed to leave piles of apples wherever he wants and pick them up later. Help Johnny figure out how many kilometers he can travel.

Input

The input consists of a single line with two space separated integers N and K.

Output

Output the number of kilometers he can travel down the road

Limits

- $0 \leq N \leq 1\,000\,000$
- $1 \leq K \leq 1\,000\,000$

Example explanation

One way of solving the first example input would be to pick up 4 apples, go to right before the 3 kilometer mark, dump the remaining two apples (having paid one at each of the toll booths at kilometer mark 1 and 2). Then go back to the pile and pick up the rest of the apples, and do the same again. This leaves him at 3 kilometers with 4 apples. He can then go an additional 4 kilometers, paying at the 3, 4, 5, and 6 kilometer marks. He will be stopped at the 7 kilometer mark because he has no apples left.

Sample Input 1	Sample Output 1
8 4	7

What's In It For Me?
Problem ID: whatsinit

Pekka has started preparing for the beautiful summer on the beaches along the great Finnish lakes. He has started his yearly spring diet. Although the diet is very strict and boring, it has given Pekka fruitful results for the last 10 years as well as an increased interest in the food he eats. He always reads the ingredient list of a food item before he puts it into his basket in the grocery store. He has a hard time getting a full overview of how much of each ingredient he actually puts in his body and would like your help to figure it out.

The ingredient list on a food item always lists all the ingredients, but it seldom tells you how much is used of each one of the single ingredients. It will often tell you how much is used for some of the ingredients. The ingredients list is always sorted from the ingredient with the most content to the ingredient with the least content.

Input

The first line of the input consists of a single integer N indicating the number of ingredients in the thing Pekka wants to eat.
Each of the next N lines describe one ingredient, and consist of one word I and one number P, describing the name of the ingredient and the percentage of the ingredient in this piece of food – if known, otherwise it is a '?'.
The N ingredients are listed in the order on the package of the food item.

Output

For each test case, output a list of ingredients, one line for each ingredient, in the same order as they were given in the input. Each line should consist of an ingredient name I, and two non-negative integers L and H, describing the minimal and maximal possible quantity of the ingredient in the actual food item.

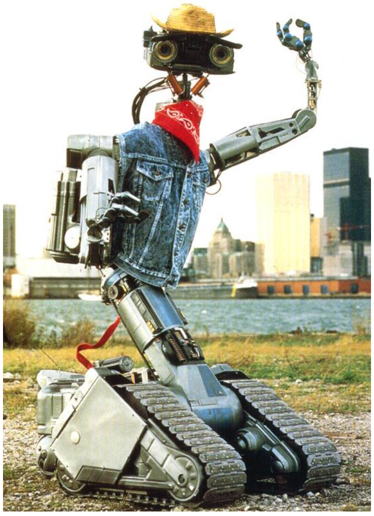
Limits

- $1 \leq N \leq 1000$
- $1 \leq \text{len}(I) \leq 100$
- P is an integer if the percentage of this ingredient is known, or a ? if it is not known
- If P is known: $0 \leq P \leq 100$
- $0 \leq \sum(P) \leq 100$
- I consists of the letters from A-Z and a-z.
- Each ingredient name I is unique
- It is guaranteed that there exists a solution

Sample Input 1	Sample Output 1
4 Strawberries 60 Sugar ? Water 10 Thickener ?	Strawberries 60 60 Sugar 20 30 Water 10 10 Thickener 0 10

Johnny5 And The Exploding Oil Cans

Problem ID: johnny5



In the hit video game "Johnny5 and the exploding oil cans", you control the robot "Johnny5". You control the robot by moving it one cell at a time in one of the directions up, down, left, or right on an $N \times N$ grid. One move takes one second and costs one unit of energy. If you have no energy, you can't move. Johnny starts the game with E units of energy.

The objective of the game is to collect as many oil cans as you can. Every second there may appear one or more oil cans somewhere on the grid. If Johnny5 is there at that point in time he collects these cans, otherwise they explode. You score 1 point for each oil can you collect. If an oil can explodes in one of the four adjacent cells to where Johnny5 is located he collects the spilled oil, and gains one unit of energy for each of them. If he does not pick up the can, and is not in one of the adjacent cells to pick up the oil, the oil disappears immediately. Note that he only gets oil from adjacent cells, and not from any cans in the same cell that he is in.

You've had trouble beating this game for years, but your friend just called and told you there is a way to get a list of where and when the cans will appear. Write a program that uses this information to find the maximum number of points you can get.

Input

The first line of the input consists of 5 space separated integers N, E, S_X, S_Y, C. These numbers give the size of the grid, the starting energy, the x and y coordinates where Johnny5 starts, and the number of cans.

The next C lines each consist of 3 space separated integers X, Y, CT. These numbers represent the x and y coordinates of a can, and the time it appears there, in seconds after the start of the game.

Output

Output the maximum number of points you can score.

Limits

- $1 \leq N \leq 500$
- $0 \leq E \leq 100$
- $0 \leq C \leq 100$
- $0 \leq X_S, Y_S, X, Y < N$
- $1 \leq CT \leq 100$

Sample Input 1	Sample Output 1
3 1 0 0 2 1 2 2 1 1 1	0

Sample Input 2

```
3 1 1 1 8
0 1 1
1 0 1
2 1 1
1 2 1
1 2 2
2 2 3
0 2 5
1 2 6
```

Sample Output 2

```
4
```

Sample Input 3

```
3 1 0 0 1
1 0 100
```

Sample Output 3

```
1
```

Saving For Retirement
Problem ID: savingforretirement

Alice is saving for her retirement. She hasn't really decided how much she wants to save, but when she retires, she wants to have strictly more money than Bob will have when he retires.

Bob is B years old. He plans to retire when he becomes B_r years old. He saves B_s every year from now until then.

Alice is A years old. She wants to save A_s every year. When is the earliest time she can retire?

Input

The input is a single line consisting of 5 space separated integers; B, B_r, B_s, A, A_s.

Output

Output the age at which Alice can retire so that she has more money than Bob will have at age B_r.

Limits

- $20 \leq B \leq B_r \leq 100$
- $20 \leq A \leq 100$
- $1 \leq A_s, B_s, 100$

Explanation of first example

At the age of 25 Bob has saved 5 every year for 5 years. This means he has 25 saved up.
At the age of 23 Alice has saved 10 every year for 3 years. This means she has 30 saved up, which is strictly more than 25.

Sample Input 1	Sample Output 1
`20 25 5 20 10`	`23`

Sample Input 2	Sample Output 2
`20 28 5 30 9`	`35`

Square Peg in a Round Hole

Problem ID: squarepegs

Mr. Johnson likes to build houses. In fact, he likes it so much that he has built a lot of houses that he has not yet placed on plots. He has recently acquired N circular plots. The city government has decided that there can be only one house on each plot, and a house cannot touch the boundary of the plot.

Mr. Johnson has M circular houses and K square houses. Help him figure out how many of the plots he can fill with houses so that he can get some money back on his investments.

Input

The first line of the input consists of 3 space separated integers; N, M, K

The next line contains N space separated integers, where the i^{th} integer denote the radius r_i of the i^{th} plot.

The next line contains M space separated integers, where the i^{th} integer denote the radius r_i of the i^{th} circular house.

The next line contains K space separated integers, where the i^{th} integer denote the side length s_i of the i^{th} square house.

Output

Output the largest number of plots he can fill with houses.

Limits

- $1 \leq N, M, K, r_i, s_i \leq 100$

Sample Input 1	Sample Output 1
5 3 3 1 2 6 7 8 2 6 7 4 8 9	3

Fencing Lessons
Problem ID: fencing

Donald's garden has N trees and M poles that he has erected in strategic locations on integer meter coordinates. He wants to start breeding squirrels on top of these poles. Unfortunately there are lots of dangerous predators in the area, so if he leaves the poles unprotected, they will eat all of the squirrels. In order to keep the squirrels safe, he has decided to build a single, connected fence around the poles. As we all know, it is not possible to breed squirrels if all the poles are placed on a single line, so Donald has made sure to avoid this.

Building a fence is hard work, so he wants to build the fence as short as possible, while still completely enclosing all the poles. The poles are so thin that they can be considered points.

In order to build the fence, he has to cut down some trees and make boards to build it from. For each of the trees in the garden, he knows both how many meters (m) of boards he gets, and how long it will take him to do it (t). Donald wants to start breeding squirrels as soon as possible, so he is also in a hurry. Help Donald figure out which trees to use to build his fence as quickly as possible.

Input

The input starts with a line with two space separated integers, N and M.
The next N lines each consists of two space separated integers m_i and t_i describing the i^{th} tree.
The next M lines each consists of two space separated integers x_i and y_i describing the coordinates of the i^{th} pole's position.

Output

Output the minimum amount of time it takes to produce the boards necessary for the fence.

Limits

- $1 \leq N \leq 1\,000$
- $3 \leq M \leq 1\,000$
- $1 \leq m_i, t_i \leq 1\,000$
- $0 \leq x_i, y_i \leq 1\,000$
- $\sum_{i=1}^{N} m_i$ will always be large enough to build the fence
- No two poles will have the same coordinates
- The length of the fence will not be within 10^{-6} of an integer
- The positions of the poles will not all be colinear

Sample Input 1	Sample Output 1
3 3 4 10 2 4 2 4 0 0 0 1 1 0	8