

Problem Set: Day 2, Summer Camp 2009

Japanese Alumni Group

Contest Held on 19 Sep 2009

This problem set is provided under CC0 1.0 Universal. The statement is available at:

<http://creativecommons.org/publicdomain/zero/1.0/>

In summary, this problem set can be used as if it were in public domain and is provided without any kind of warranties.

Problem A

Infected Computer

Summer Camp (Day 2), Tokyo
19 Sep 2009

Adam Ivan is working as a system administrator at Soy Group, Inc. He is now facing a big trouble: a number of computers under his management have been infected by a computer virus. Unfortunately, anti-virus system in his company failed to detect this virus because it was very new.

Adam has identified the first computer infected by the virus and collected the records of all data packets sent within his network. He is now trying to identify which computers have been infected. A computer is infected when receiving any data packet from any infected computer. The computer is not infected, on the other hand, just by sending data packets to infected computers.

It seems almost impossible for him to list all infected computers by hand, because the size of the packet records is fairly large. So he asked you for help: write a program that can identify infected computers.

► Input

The input has the following format:

```
 $N$   $M$   
 $t_1$   $s_1$   $d_1$   
 $t_2$   $s_2$   $d_2$   
...  
 $t_M$   $s_M$   $d_M$ 
```

N is the number of computers; M is the number of data packets; t_i ($1 \leq i \leq M$) is the time when the i -th data packet is sent; s_i and d_i ($1 \leq i \leq M$) are the source and destination computers of the i -th data packet respectively. The first infected computer is indicated by the number 1; the other computers are indicated by unique numbers between 2 and N .

The input meets the following constraints: $0 < N \leq 20000$, $0 \leq M \leq 20000$, and $0 \leq t_i \leq 10^9$ for $1 \leq i \leq M$; all t_i 's are different; and the source and destination of each packet are always different.

► Output

Print the number of computers infected by the computer virus.

► Sample Input and Output

```
Input #1:  
3 2  
1 1 2  
2 2 3
```

```
Output #1:  
3
```

```
Input #2:  
3 2  
2 3 2  
1 2 1
```

```
Output #2:  
1
```

Problem B Magic Slayer

Summer Camp (Day 2), Tokyo
19 Sep 2009

You are in a fantasy monster-ridden world. You are a slayer fighting against the monsters with magic spells.

The monsters have *hit points* for each, which represent their vitality. You can decrease their hit points by your magic spells: each spell gives certain points of *damage*, by which monsters lose their hit points, to either one monster or all monsters in front of you (depending on the spell). Monsters are defeated when their hit points decrease to less than or equal to zero. On the other hand, each spell may consume a certain amount of your *magic power*. Since your magic power is limited, you want to defeat monsters using the power as little as possible.

Write a program for this purpose.

► Input

The input is given in the following format:

```
N  
HP1  
HP2  
...  
HPN  
M  
Name1 MP1 Target1 Damage1  
Name2 MP2 Target2 Damage2  
...  
NameM MPM TargetM DamageM
```

N is the number of monsters in front of you ($1 \leq N \leq 100$); HP_i is the hit points of the i -th monster ($1 \leq HP_i \leq 100000$); M is the number of available magic spells ($1 \leq M \leq 100$); $Name_j$ is the name of the j -th spell, consisting of up to 16 uppercase and lowercase letters; MP_j is the amount of magic power consumed by the j -th spell ($0 \leq MP_j \leq 99$); $Target_j$ is either "Single" or "All", where these indicate the j -th magic gives damage just to a single monster or to all monsters respectively; $Damage_j$ is the amount of damage (per monster in case of "All") made by the j -th magic ($0 \leq Damage_j \leq 999999$).

All the numbers in the input are integers. There is at least one spell that gives non-zero damage to monsters.

► Output

Print in a line the minimum amount of magic power consumed to defeat all the monsters in the input.

► Sample Input and Output

Input #1:

3

8000 15000 30000

3

Flare 45 Single 8000

Meteor 62 All 6000

Ultimate 80 All 9999

Output #1:

232

Problem C

Dial Lock

Summer Camp (Day 2), Tokyo
19 Sep 2009

A dial lock is a kind of lock which has some dials with printed numbers. It has a special sequence of numbers, namely an unlocking sequence, to be opened.

You are working at a manufacturer of dial locks. Your job is to verify that every manufactured lock is unlocked by its unlocking sequence. In other words, you have to rotate many dials of many many locks. It's a very hard and boring task. You want to reduce time to open the locks.

It's a good idea to rotate multiple dials at one time. It is, however, a difficult problem to find steps to open a given lock with the fewest rotations. So you decided to write a program to find such steps for given initial and unlocking sequences.

Your company's dial locks are composed of vertically stacked k ($1 \leq k \leq 10$) cylindrical dials. Every dial has 10 numbers, from 0 to 9, along the side of the cylindrical shape from the left to the right in sequence. The neighbor on the right of 9 is 0.

A dial points one number at a certain position. If you rotate a dial to the left by i digits, the dial newly points the i -th right number. In contrast, if you rotate a dial to the right by i digits, it points the i -th left number. For example, if you rotate a dial pointing 8 to the left by 3 digits, the dial newly points 1.

You can rotate more than one adjacent dial at one time. For example, consider a lock with 5 dials. You can rotate just the 2nd dial. You can rotate the 3rd, 4th and 5th dials at the same time. But you cannot rotate the 1st and 3rd dials at one time without rotating the 2nd dial. When you rotate multiple dials, you have to rotate them to the same direction by the same digits.

Your program is to calculate the fewest number of rotations to unlock, for given initial and unlocking sequences. Rotating one or more adjacent dials to the same direction by the same digits is counted as one rotation.

► Input

The input consists of two lines. The first line contains an integer k . The second lines contain two strings, separated by a space, which indicate the initial and unlocking sequences.

► Output

Print the minimum number of rotations in one line.

► Sample Input and Output

Input #1:
4
1357 4680

Output #1:
1

Input #2:
6
777777 003330

Output #2:
2

Problem D

Double Sorting

Summer Camp (Day 2), Tokyo
19 Sep 2009

Here we describe a typical problem. There are n balls and n boxes. Each ball is labeled by a unique number from 1 to n . Initially each box contains one of these balls. We can swap two balls in adjacent boxes. We are to sort these balls in increasing order by swaps, i.e. move the ball labeled by 1 to the first box, labeled by 2 to the second box, and so forth. The question is how many swaps are needed.

Now let us consider the situation where the balls are doubled, that is, there are $2n$ balls and n boxes, exactly two balls are labeled by k for each $1 \leq k \leq n$, and the boxes contain two balls each. We can swap two balls in adjacent boxes, one ball from each box. We are to move the both balls labeled by 1 to the first box, labeled by 2 to the second box, and so forth. The question is again how many swaps are needed.

Here is one interesting fact. We need 10 swaps to sort [5; 4; 3; 2; 1] (the state with 5 in the first box, 4 in the second box, and so forth): swapping 5 and 4, then 5 and 3, 5 and 2, 5 and 1, 4 and 3, 4 and 2, 4 and 1, 3 and 2, 3 and 1, and finally 2 and 1. Then how many swaps we need to sort [5, 5; 4, 4; 3, 3; 2, 2; 1, 1] (the state with two 5's in the first box, two 4's in the second box, and so forth)? Some of you might think 20 swaps — this is not true, but the actual number is 15.

Write a program that calculates the number of swaps for the two-ball version and verify the above fact.

► Input

The input has the following format:

```
 $n$   
 $ball_{1,1}$   $ball_{1,2}$   
 $ball_{2,1}$   $ball_{2,2}$   
...  
 $ball_{n,1}$   $ball_{n,2}$ 
```

n is the number of boxes ($1 \leq n \leq 8$). $ball_{i,1}$ and $ball_{i,2}$, for $1 \leq i \leq n$, are the labels of two balls initially contained by the i -th box.

► Output

Print the minimum possible number of swaps.

► Sample Input and Output

Input #1:

```
5  
5 5  
4 4  
3 3  
2 2  
1 1
```

Output #1:

```
15
```

Input #2:

5

1 5

3 4

2 5

2 3

1 4

Output #2:

9

Input #3:

8

8 3

4 2

6 4

3 5

5 8

7 1

2 6

1 7

Output #3:

21

Problem E

Symmetry

Summer Camp (Day 2), Tokyo
19 Sep 2009

Open Binary and Object Group organizes a programming contest every year. Mr. Hex belongs to this group and joins the judge team of the contest. This year, he created a geometric problem with its solution for the contest. The problem required a set of points forming a line-symmetric polygon for the input. Preparing the input for this problem was also his task. The input was expected to cover all edge cases, so he spent much time and attention to make them satisfactory.

However, since he worked with lots of care and for a long time, he got tired before he finished. So He might have made mistakes — there might be polygons not meeting the condition. It was not reasonable to prepare the input again from scratch. The judge team thus decided to find all line-asymmetric polygons in his input and fix them as soon as possible. They asked a programmer, just you, to write a program to find incorrect polygons.

You can assume the following:

- Edges of the polygon must not cross or touch each other except for the end points of adjacent edges.
- It is acceptable for the polygon to have adjacent three vertexes on a line, but in such a case, there must be the vertex symmetric to each of them.

► Input

The input consists of a set of points in the following format.

```
N
x1 y1
x2 y2
...
xN yN
```

The first line of the input contains an integer N ($3 \leq N \leq 1000$), which denotes the number of points. The following N lines describe each point. The i -th line contains two integers x_i, y_i ($-10000 \leq x_i, y_i \leq 10000$), which denote the coordinates of the i -th point.

Note that, although the points are the vertexes of a polygon, they are given in an arbitrary order, not necessarily clockwise or counterclockwise.

► Output

Output “Yes” in a line if the points can form a line-symmetric polygon, otherwise output “No”.

► Sample Input and Output

Input #1:	Output #1:
4	Yes
0 1	
1 0	
0 0	
1 1	

Input #2:
4
0 1
1 -1
0 0
1 1

Output #2:
No

Input #3:
9
-1 1
0 1
1 1
-1 0
0 0
1 0
-1 -1
0 -1
1 -1

Output #3:
No

Input #4:
3
-1 -1
0 0
1 1

Output #4:
No

Input #5:
4
0 2
0 0
-1 0
1 0

Output #5:
Yes

Problem F Voronoi Island

Summer Camp (Day 2), Tokyo
19 Sep 2009

Some of you know an old story of Voronoi Island. There were N liege lords and they are always involved in territorial disputes. The residents of the island were despaired of the disputes.

One day, a clever lord proposed to stop the disputes and divide the island fairly. His idea was to divide the island such that any piece of area belongs to the lord of the nearest castle. The method is called Voronoi Division today.

Actually, there are many aspects of whether the method is fair. According to one historian, the clever lord suggested the method because he could gain broader area than other lords.

Your task is to write a program to calculate the size of the area each lord can gain. You may assume that Voronoi Island has a convex shape.

► Input

The input has the following format:

```
 $N$   $M$   
 $Ix_1$   $Iy_1$   
 $Ix_2$   $Iy_2$   
...  
 $Ix_N$   $Iy_N$   
 $Cx_1$   $Cy_1$   
 $Cx_2$   $Cy_2$   
...  
 $Cx_M$   $Cy_M$ 
```

N is the number of the vertices of Voronoi Island; M is the number of the liege lords; (Ix_i, Iy_i) denotes the coordinate of the i -th vertex; (Cx_i, Cy_i) denotes the coordinate of the castle of the i -th lord.

The input meets the following constraints: $3 \leq N \leq 10$, $2 \leq M \leq 10$, $-100 \leq Ix_i, Iy_i, Cx_i, Cy_i \leq 100$. The vertices are given counterclockwise. All the coordinates are integers.

► Output

Print the area gained by each liege lord with an absolute error of at most 10^{-4} . You may output any number of digits after the decimal point. The order of the output areas must match that of the lords in the input.

► Sample Input and Output

Input #1:

3 3
0 0
8 0
0 8
2 2
4 2
2 4

Output #1:

9.0
11.5
11.5

Problem G

Defend the Bases

Summer Camp (Day 2), Tokyo
19 Sep 2009

A country Gizevom is being under a sneak and fierce attack by their foe. They have to deploy one or more troops to every base immediately in order to defend their country. Otherwise their foe would take all the bases and declare "All your base are belong to us."

You are asked to write a program that calculates the minimum time required for deployment, given the present positions and marching speeds of troops and the positions of the bases.

► Input

The input is given in the following format:

```
N M
x1 y1 v1
x2 y2 v2
...
xN yN vN
x'1 y'1
x'2 y'2
...
x'M y'M
```

N is the number of troops ($1 \leq N \leq 100$); M is the number of bases ($1 \leq M \leq 100$); (x_i, y_i) denotes the present position of i -th troop; v_i is the speed of the i -th troop ($1 \leq v_i \leq 100$); (x'_j, y'_j) is the position of the j -th base.

All the coordinates are integers between 0 and 10000 inclusive.

► Output

Print the minimum required time in a line.

► Sample Input and Output

Input #1:

```
2 2
10 20 1
0 10 1
0 10
10 0
```

Output #1:

```
14.14213562
```

Problem H

Galaxy Wide Web Service

Summer Camp (Day 2), Tokyo
19 Sep 2009

The volume of access to a web service varies from time to time in a day. Also, the hours with the highest volume of access varies from service to service. For example, a service popular in the United States may receive more access in the daytime in the United States, while another service popular in Japan may receive more access in the daytime in Japan. When you develop a web service, you have to design the system so it can handle all requests made during the busiest hours.

You are a lead engineer in charge of a web service in the 30th century. It's the era of Galaxy Wide Web (GWW), thanks to the invention of faster-than-light communication. The service can be accessed from all over the galaxy. Thus many intelligent creatures, not limited to human beings, can use the service. Since the volume of access to your service is increasing these days, you have decided to reinforce the server system. You want to design a new system that handles requests well even during the hours with the highest volume of access. However, this is not a trivial task. Residents in each planet have their specific length of *a day*, say, a cycle of life. The length of *a day* is not always 24 hours. Therefore, a cycle of the volume of access are different by planets of users.

You have obtained hourly data of the volume of access for all planets where you provide the service. Assuming the volume of access follows a daily cycle for each planet, you want to know the highest volume of access in one hour. It should be a quite easy task for you, a famous talented engineer in the galaxy.

► Input

The input is formatted as follows.

$$\begin{array}{l} N \\ d_1 \ t_1 \ q_{1,0} \ \dots \ q_{1,d_1-1} \\ \dots \\ d_N \ t_N \ q_{N,0} \ \dots \ q_{N,d_N-1} \end{array}$$

N is the number of planets. d_i ($1 \leq i \leq N$) is the length of a day in the planet i . $q_{i,j}$ is the volume of access on the planet i during from the j -th hour to the $(j + 1)$ -th hour.

You may assume that $N \leq 100$, $d_i \leq 24$, $q_{i,j} \leq 1000000$ ($1 \leq i \leq N$, $0 \leq j \leq d_i - 1$).

► Output

Output the maximum volume of access in one hour in a line.

► Sample Input and Output

Input #1:	Output #1:
2	5
4 0 1 2 3 4	
2 0 2 1	

Problem I

Tatami

Summer Camp (Day 2), Tokyo
19 Sep 2009

A tatami mat, a Japanese traditional floor cover, has a rectangular form with aspect ratio 1:2. When spreading tatami mats on a floor, it is prohibited to make a cross with the border of the tatami mats, because it is believed to bring bad luck.

Your task is to write a program that reports how many possible ways to spread tatami mats of the same size on a floor of given height and width.

► Input

The input consists of a line which contains two integers H and W in this order, separated with a single space. H and W are the height and the width of the floor respectively. The length of the shorter edge of a tatami mat is regarded as a unit length.

You may assume $0 < H, W \leq 20$.

► Output

Print the number of possible ways to spread tatami mats in one line.

► Sample Input and Output

Input #1:

3 4

Output #1:

4

Input #2:

4 4

Output #2:

2

Problem J

Revenge of the Round Table

Summer Camp (Day 2), Tokyo
19 Sep 2009

Two countries A and B have decided to make a meeting to get acquainted with each other. n ambassadors from A and B will attend the meeting in total.

A round table is prepared for in the meeting. The ambassadors are getting seated at the round table, but they have agreed that more than k ambassadors from the same country does not sit down at the round table in a row for deeper exchange.

Your task is to write a program that reports the number of possible arrangements when rotations are not counted. Your program should report the number modulo $M = 1000003$.

Let us provide an example. Suppose $n = 4$ and $k = 2$. When rotations are counted as different arrangements, the following six arrangements are possible.

AABB
ABBA
BBAA
BAAB
ABAB
BABA

However, when rotations are regarded as same, the following two arrangements are possible.

AABB
ABAB

Therefore the program should report 2.

► Input

The input consists of two integers n ($1 \leq n \leq 1000$) and k ($1 \leq k \leq 1000$) in one line.

It does not always hold $k < n$. This means the program should consider cases in which the ambassadors from only one country attend the meeting.

► Output

Print the number of possible arrangements modulo $M = 1000003$ in one line.

► Sample Input and Output

Input #1: 3 1	Output #1: 0
Input #2: 3 2	Output #2: 2
Input #3: 3 3	Output #3: 4

Input #4:
4 2

Output #4:
2

Input #5:
10 5

Output #5:
90

Input #6:
1000 500

Output #6:
570682