Problem Set: Day 3, Summer Camp 2009

Japanese Alumni Group

Contest Held on 20 Sep 2009

Problem A
Strange String Manipulation

Summer Camp (Day 3), Tokyo
20 Sep 2009

A linear congruential generator produces a series $R(\cdot)$ of pseudo-random numbers by the following formulas:

$$R(0) = S, R(i) = (A \cdot R(i-1) + C) \bmod M \quad (\text{for } i = 1, 2, \ldots),$$

where S, A, C, and M are all parameters. In this problem, $0 \le S, A, C \le 15$ and $M = 256$.

Now suppose we have some input string $I(\cdot)$, where each character in the string is an integer between 0 and $(M-1)$. Then, using the pseudo-random number series $R(\cdot)$, we obtain another string $O(\cdot)$ as the output by the following formula:

$$O(i) = (I(i) + R(i)) \bmod M \quad (\text{for } i = 1, 2, \ldots),$$

Your task is to write a program that shows the parameters S, A, and C such that the information entropy of the output string $O(\cdot)$ is minimized. Here, the information entropy H is given by the following formula:

$$H = -\sum_x \frac{\#(x)}{N} \log \frac{\#(x)}{N}$$

where N is the length of the string and $\#(x)$ is the number of occurences of the alphabet x.

▶ Input

The input has the following format:

> N
> $I(1)\ I(2) \ldots I(N)$

N does not exceed 256.

▶ Output

Print in a line the values of the three parameters S, A, and C separated by a single space. If more than one solution gives the same minimum entropy, choose the solution with the smallest S, A, and then C.

▶ Sample Input and Output

Input #1:
```
5
5 4 3 2 1
```

Output #1:
```
0 1 1
```

Input #2:
```
5
7 7 7 7 7
```

Output #2:
```
0 0 0
```

Input #3:

```
10
186 8 42 24 154 40 10 56 122 72
```

Output #3:

```
8 7 14
```

Problem B
Erratic Sleep Habits

Summer Camp (Day 3), Tokyo
20 Sep 2009

Peter is a person with erratic sleep habits. He goes to sleep at twelve o'clock every midnight. He gets up just after one hour of sleep on some days; he may even sleep for twenty-three hours on other days. His sleeping duration changes in a cycle, where he always sleeps for only one hour on the first day of the cycle.

Unfortunately, he has some job interviews this month. No doubt he wants to be in time for them. He can take anhydrous caffeine to reset his sleeping cycle to the beginning of the cycle anytime. Then he will wake up after one hour of sleep when he takes caffeine. But of course he wants to avoid taking caffeine as possible, since it may affect his health and accordingly his very important job interviews.

Your task is to write a program that reports the minimum amount of caffeine required for Peter to attend all his interviews without being late, where the information about the cycle and the schedule of his interviews are given. The time for move to the place of each interview should be considered negligible.

► Input

The input is given in the following format:

T
$t_1\ t_2\ \ldots\ t_T$
N
$D_1\ M_1$
$D_2\ M_2$
$\ldots$
$D_N\ M_N$

T is the length of the cycle ($1 \le T \le 30$); t_i (for $1 \le i \le T$) is the amount of sleep on the i-th day of the cycle, given in hours ($1 \le t_i \le 23$); N is the number of interviews ($1 \le N \le 100$); D_j (for $1 \le j \le N$) is the day of the j-th interview ($1 \le D_j \le 100$); M_j (for $1 \le j \le N$) is the hour when the j-th interview starts ($1 \le M_j \le 23$).

The numbers in the input are all integers. t_1 is always 1 as stated above. The day indicated by 1 is the first day in the cycle of Peter's sleep.

► Output

Print the minimum number of times Peter needs to take anhydrous caffeine.

► Sample Input and Output

Input #1:
```
2
1 23
3
1 1
2 1
3 1
```

Output #1:
```
2
```

Problem C
Find the Point

Summer Camp (Day 3), Tokyo
20 Sep 2009

We understand that reading English is a great pain to many of you. So we'll keep this problem statememt simple. Write a program that reports the point equally distant from a set of lines given as the input. In case of no solutions or multiple solutions, your program should report as such.

► Input

The input is given in the following format:

n
$x_{1,1}\ y_{1,1}\ x_{1,2}\ y_{1,2}$
$x_{2,1}\ y_{2,1}\ x_{2,2}\ y_{2,2}$
...
$x_{n,1}\ y_{n,1}\ x_{n,2}\ y_{n,2}$

n is the number of lines ($1 \leq n \leq 100$); $(x_{i,1}, y_{i,1})$ and $(x_{i,2}, y_{i,2})$ denote the different points the i-th line passes through. The lines do not coincide each other. The coordinates are all integers between -10000 and 10000.

► Output

Print a line as follows. If there is exactly one point equally distant from all the given lines, print the x- and y-coordinates in this order with a single space between them. If there is more than one such point, just print "`Many`" (without quotes). If there is none, just print "`None`" (without quotes).

The coordinates may be printed with any number of digits after the decimal point, but should be accurate to 10^{-4}.

► Sample Input and Output

Input #1:

```
2
-35 -35 100 100
-49 49 2000 -2000
```

Output #1:

```
Many
```

Input #2:

```
4
0 0 0 3
0 0 3 0
0 3 3 3
3 0 3 3
```

Output #2:

```
1.5000 1.5000
```

Input #3:

```
4
0 3 -4 6
3 0 6 -4
2 3 6 6
-1 2 -4 6
```

Output #3:

```
1.0000 1.0000
```

Problem D
Luigi's Tavern

Summer Camp (Day 3), Tokyo
20 Sep 2009

Luigi's Tavern is a thriving tavern in the Kingdom of Nahaila. The owner of the tavern Luigi supports to organize a party, because the main customers of the tavern are adventurers. Each adventurer has a job: hero, warrior, cleric or mage.

Any party should meet the following conditions:

- A party should have a hero.
- The warrior and the hero in a party should get along with each other.
- The cleric and the warrior in a party should get along with each other.
- The mage and the cleric in a party should get along with each other.
- It is recommended that a party has a warrior, a cleric, and a mage, but it is allowed that at most N_W, N_C and N_m parties does not have a warrior, a cleric, and a mage respectively.
- A party without a cleric should have a warrior and a mage.

Now, the tavern has H heroes, W warriors, C clerics and M mages. Your job is to write a program to find the maximum number of parties they can form.

► Input

The first line of the input contains 7 non-negative integers H, W, C, M, N_W, N_C, and N_M, each of which is less than or equals to 50. The i-th of the following W lines contains the list of heroes who will be getting along with the warrior i. The list begins with a non-negative integer n_i, less than or equals to H. Then the rest of the line should contain n_i positive integers, each of which indicates the ID of a hero getting along with the warrior i.

After these lists, the following C lines contain the lists of warriors getting along with the clerics in the same manner. The j-th line contains a list of warriors who will be getting along with the cleric j. Then the last M lines of the input contain the lists of clerics getting along with the mages, of course in the same manner. The k-th line contains a list of clerics who will be getting along with the mage k.

► Output

You should output the maximum number of parties possible.

► Sample Input and Output

Input #1:

```
2 1 1 1 1 1 1
1 1
1 1
1 1
```

Output #1:

```
2
```

Input #2:

```
1 1 1 1 0 0 0
1 1
1 1
1 1
```

Output #2:

```
1
```

Input #3:

```
1 0 1 0 1 0 1
0
```

Output #3:

```
1
```

Input #4:

```
1 1 0 1 0 1 0
0
0
```

Output #4:

```
0
```

Input #5:

```
1 1 0 1 0 1 0
1 1
0
```

Output #5:

```
1
```

Problem E
Colored Octahedra

Summer Camp (Day 3), Tokyo
20 Sep 2009

A young boy John is playing with eight triangular panels. These panels are all regular triangles of the same size, each painted in a single color; John is forming various octahedra with them.

While he enjoys his playing, his father is wondering how many octahedra can be made of these panels since he is a pseudo-mathematician. Your task is to help his father: write a program that reports the number of possible octahedra for given panels. Here, a pair of octahedra should be considered identical when they have the same combination of the colors allowing rotation.

► Input

The input has the following format:

$$Color_1 \; Color_2 \ldots Color_8$$

Each $Color_i$ ($1 \leq i \leq 8$) is a string of up to 20 lowercase alphabets and represents the color of the i-th triangular panel.

► Output

Output the number of different octahedra that can be made of given panels.

► Sample Input and Output

Input #1:
```
blue blue blue blue blue blue blue blue
```
Output #1:
```
1
```

Input #2:
```
red blue blue blue blue blue blue blue
```
Output #2:
```
1
```

Input #3:
```
red red blue blue blue blue blue blue
```
Output #3:
```
3
```

Problem F
Marked Ancestor

Summer Camp (Day 3), Tokyo
20 Sep 2009

You are given a tree T that consists of N nodes. Each node is numbered from 1 to N, and node 1 is always the root node of T. Consider the following two operations on T:

- `M` v: (Mark) Mark node v.
- `Q` v: (Query) Print the index of the nearest marked ancestor of node v which is nearest to it. Initially, only the root node is marked.

Your job is to write a program that performs a sequence of these operations on a given tree and calculates the value that each `Q` operation will print. To avoid too large output file, your program is requested to print the sum of the outputs of all query operations. Note that the judges confirmed that it is possible to calculate every output of query operations in a given sequence.

► Input

The first line of the input contains two integers N and Q, which denotes the number of nodes in the tree T and the number of operations, respectively. These numbers meet the following conditions: $1 \le N \le 100000$ and $1 \le Q \le 100000$.

The following $N - 1$ lines describe the configuration of the tree T. Each line contains a single integer p_i ($i = 2, \ldots, N$), which represents the index of the parent of i-th node.

The next Q lines contain operations in order. Each operation is formatted as "`M` v" or "`Q` v", where v is the index of a node.

► Output

Print the sum of the outputs of all query operations in one line.

► Sample Input and Output

Input #1:

```
6 3
1
1
2
3
3
Q 5
M 3
Q 5
```

Output #1:

```
4
```

Problem G
Strange Couple

Summer Camp (Day 3), Tokyo
20 Sep 2009

Alice and Bob are going to drive from their home to a theater for a date. They are very challenging — they have no maps with them even though they don't know the route at all (since they have just moved to their new home). Yes, they will be going just by their feeling.

The town they drive can be considered as an undirected graph with a number of intersections (vertices) and roads (edges). Each intersection may or may not have a sign. On intersections with signs, Alice and Bob will enter the road for the shortest route. When there is more than one such roads, they will go into one of them at random.

On intersections without signs, they will just make a random choice. Each random selection is made with equal probabilities. They can even choose to go back to the road they have just come along, on a random selection.

Calculate the expected distance Alice and Bob will drive before reaching the theater.

► Input

The input has the following format:

$$
\begin{array}{l}
n\ s\ t \\
q_1\ q_2\ \dots\ q_n \\
a_{11}\ a_{12}\ \dots\ a_{1n} \\
a_{21}\ a_{22}\ \dots\ a_{2n} \\
\vdots \\
a_{n1}\ a_{n2}\ \dots\ a_{nn}
\end{array}
$$

n is the number of intersections ($n \leq 100$). s and t are the intersections the home and the theater are located respectively ($1 \leq s, t \leq n$, $s \neq t$); q_i (for $1 \leq i \leq n$) is either 1 or 0, where 1 denotes there is a sign at the i-th intersection and 0 denotes there is not; a_{ij} (for $1 \leq i, j \leq n$) is a positive integer denoting the distance of the road connecting the i-th and j-th intersections, or 0 indicating there is no road directly connecting the intersections. The distance of each road does not exceed 10.

Since the graph is undirectional, it holds $a_{ij} = a_{ji}$ for any $1 \leq i, j \leq n$. There can be roads connecting the same intersection, that is, it does *not* always hold $a_{ii} = 0$. Also, note that the graph is not always planar.

► Output

Print the expected distance accurate to 10^{-8}, or "`impossible`" (without quotes) if there is no route to get to the theater. The distance may be printed with any number of digits after the decimal point.

► Sample Input and Output

Input #1:

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

Output #1:

```
11.00000000
```

Problem H
Queen's Case

Summer Camp (Day 3), Tokyo
20 Sep 2009

A small country called Maltius was governed by a queen. The queen was known as an oppressive ruler. People in the country suffered from heavy taxes and forced labor. So some young people decided to form a revolutionary army and fight against the queen. Now, they besieged the palace and have just rushed into the entrance.

Your task is to write a program to determine whether the queen can escape or will be caught by the army.

Here is detailed description.

- The palace can be considered as grid squares.
- The queen and the army move alternately. The queen moves first.
- At each of their turns, they either move to an adjacent cell or stay at the same cell.
- Each of them must follow the optimal strategy.
- If the queen and the army are at the same cell, the queen will be caught by the army immediately.
- If the queen is at any of exit cells alone after the army's turn, the queen can escape from the army.
- There may be cases in which the queen cannot escape but won't be caught by the army forever, under their optimal strategies.

► Input

The input describes a map of the palace. The first line of the input contains two integers W ($1 \leq W \leq 30$) and H ($1 \leq H \leq 30$), which indicate the width and height of the palace. The following H lines, each of which contains W characters, denote the map of the palace. "`Q`" indicates the queen,"`A`" the army,"`E`" an exit,"`#`" a wall and "`.`" a floor.

The map contains exactly one "`Q`", exactly one "`A`" and at least one "`E`". You can assume both the queen and the army can reach all the exits.

► Output

Output "`Queen can escape.`", "`Army can catch Queen.`" or "`Queen can not escape and Army can not catch Queen.`" in a line.

► Sample Input and Output

Input #1:

```
2 2
QE
EA
```

Output #1:

```
Queen can not escape and Army can not catch Queen.
```

Input #2:

```
3 1
QAE
```

Output #2:

```
Army can catch Queen.
```

Input #3:	Output #3:
`3 1` `AQE`	`Queen can escape.`

```
3 1
AQE
```

Input #3 / Output #3: `Queen can escape.`

Input #4:

```
5 5
..E..
.###.
A###Q
.###.
..E..
```

Output #4:

```
Queen can not escape and Army can not catch Queen.
```

Input #5:

```
5 1
A.E.Q
```

Output #5:

```
Army can catch Queen.
```

Input #6:

```
5 5
A....
####.
..E..
.####
....Q
```

Output #6:

```
Army can catch Queen.
```

► Hint

On the first sample input, the queen can move to exit cells, but either way the queen will be caught at the next army's turn. So the optimal strategy for the queen is staying at the same cell. Then the army can move to exit cells as well, but again either way the army will miss the queen from the other exit. So the optimal strategy for the army is also staying at the same cell. Thus the queen cannot escape but won't be caught.

Problem I
Wind Passages

Summer Camp (Day 3), Tokyo
20 Sep 2009

Wind Corridor is a covered passageway where strong wind is always blowing. It is a long corridor of width W, and there are several pillars in it. Each pillar is a right prism and its face is a polygon (not necessarily convex).

In this problem, we consider two-dimensional space where the positive x-axis points the east and the positive y-axis points the north. The passageway spans from the south to the north, and its length is infinity. Specifically, it covers the area $0 \leq x \leq W$. The outside of the passageway is filled with walls. Each pillar is expressed as a polygon, and all the pillars are located within the corridor without conflicting or touching each other.

Wind blows from the south side of the corridor to the north. For each second, w unit volume of air can be flowed at most if the minimum width of the path of the wind is w. Note that the path may fork and merge, but never overlaps with pillars and walls.

Your task in this problem is to write a program that calculates the maximum amount of air that can be flowed through the corridor per second.

► Input

The first line of the input contains two integers W and N. W is the width of the corridor, and N is the number of pillars. W and N satisfy the following condition: $2 \leq W \leq 10^4$ and $0 \leq N \leq 200$.

Then, N specifications of each pillar follow. Each specification starts with a line that contains a single integer M, which is the number of the vertices of a polygon ($3 \leq M \leq 40$). The following M lines describe the shape of the polygon. The i-th line ($1 \leq i \leq M$) contains two integers x_i and y_i that denote the coordinate of the i-th vertex ($0 < x_i < W$, $0 < y_i < 10^4$).

► Output

Your program should print a line that contains the maximum amount of air flow per second, in unit volume. The output may contain arbitrary number of digits after the decimal point, but the absolute error must not exceed 10^{-6}.

► Sample Input and Output

Input #1:

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

Output #1:

```
3.41421356
```

Problem J
Secret Operation

Summer Camp (Day 3), Tokyo
20 Sep 2009

Mary Ice is a member of a spy group. She is about to carry out a secret operation with her colleague.

She has got into a target place just now, but unfortunately the colleague has not reached there yet. She needs to hide from her enemy George Water until the colleague comes. Mary may want to make herself appear in George's sight as short as possible, so she will give less chance for George to find her.

You are requested to write a program that calculates the time Mary is in George's sight before her colleague arrives, given the information about moves of Mary and George as well as obstacles blocking their sight.

Read the Input section for the details of the situation.

▶ Input

The input has the following format:

$Time\ R$
L
$MaryX_1\ MaryY_1\ MaryT_1$
$MaryX_2\ MaryY_2\ MaryT_2$
$\ldots$
$MaryX_L\ MaryY_L\ MaryT_L$
M
$GeorgeX_1\ GeorgeY_1\ GeorgeT_1$
$GeorgeX_2\ GeorgeY_2\ GeorgeT_2$
$\ldots$
$GeorgeX_M\ GeorgeY_M\ GeorgeT_M$
N
$BlockSX_1\ BlockSY_1\ BlockTX_1\ BlockTY_1$
$BlockSX_2\ BlockSY_2\ BlockTX_2\ BlockTY_2$
$\ldots$
$BlockSX_N\ BlockSY_N\ BlockTX_N\ BlockTY_N$

The first line contains two integers. *Time* ($0 \le Time \le 100$) is the time Mary's colleague reaches the place. R ($0 < R < 30000$) is the distance George can see — he has a sight of this distance and of 45 degrees left and right from the direction he is moving. In other words, Mary is found by him if and only if she is within this distance from him and in the direction different by not greater than 45 degrees from his moving direction and there is no obstacles between them.

The description of Mary's move follows. Mary moves from $(MaryX_i, MaryY_i)$ to $(MaryX_{i+1}, MaryY_{i+1})$ straight and at a constant speed during the time between $MaryT_i$ and $MaryT_{i+1}$, for each $1 \le i \le L-1$. The following constraints apply: $2 \le L \le 20$, $MaryT_1 = 0$ and $MaryT_L = Time$, and $MaryT_i < MaryT_{i+1}$ for any $1 \le i \le L-1$.

The description of George's move is given in the same way with the same constraints, following Mary's. In addition, $(GeorgeX_j, GeorgeY_j)$ and $(GeorgeX_{j+1}, GeorgeY_{j+1})$ do not coincide for any $1 \le j \le M-1$. In other words, George is always moving in some direction.

Finally, there comes the information of the obstacles. Each obstacle has a rectangular shape occupying $(BlockSX_k, BlockSY_k)$ to $(BlockTX_k, BlockTY_k)$. No obstacle touches or crosses with another. The number of obstacles ranges from 0 to 20 inclusive.

All the coordinates are integers not greater than 10000 in their absolute values. You may assume that, if the coordinates of Mary's and George's moves would be changed within the distance of 10^{-6}, the solution would be changed by not greater than 10^{-6}.

► Output

Print the calculated time in a line. The time may be printed with any number of digits after the decimal point, but should be accurate to 10^{-4}.

► Sample Input and Output

Input #1:

```
50 100
2
50 50 0
51 51 50
2
0 0 0
1 1 50
0
```

Output #1:

```
50
```