NCPC 2012

6th October 2012

acm International Collegiate Programming Contest

IBM event sponsor

Problems

A	Aaah!
B	Bread Sorting
C	Cookie Selection
D	Doorman
E	Eco-driving
F	Food Review
G	Galactic Warlords
H	Horror List
I	Infiltration
J	Juice
K	Kindergarten
L	Luggage

Do not open before the contest has started.

Advice, hints, and general information

- Your submissions will be run multiple times, on several different input files. If your submission is incorrect, the error message you get will be the error exhibited on the first input file on which you failed. E.g., if your instance is prone to crash but also incorrect, your submission may be judged as either “wrong answer” or “run time error”, depending on which is discovered first.

- For problems with floating point output, we only require that your output is correct up to either a relative or absolute error of 10^{-6}. For example, this means that

  - If the correct answer is 0.05, any answer between 0.049999 and $.050001$ will be accepted.
  - If the correct answer is 50, any answer between 49.99995 and 50.00005 will be accepted.

  Any reasonable format for floating point numbers is acceptable. For instance, “17.000000”, “0.17e2”, and “17” are all acceptable ways of formatting the number 17. For the definition of reasonable, please use your common sense.

Problem A
Aaah!
Problem ID: aaah

Photo by Unknown

Jon Marius shouted too much at the recent Justin Bieber concert, and now needs to go to the doctor because of his sore throat. The doctor's instructions are to say "aaah". Unfortunately, the doctors sometimes need Jon Marius to say "aaah" for a while, which Jon Marius has never been good at. Each doctor requires a certain *level* of "aah" – some require "aaaaaah", while others can actually diagnose his throat with just a "h". (They often diagnose wrongly, but that is beyond the scope of this problem.) Since Jon Marius does not want to go to a doctor and have his time wasted, he wants to compare how long he manages to hold the "aaah" with the doctor's requirements. (After all, who wants to be all like "aaah" when the doctor wants you to go "aaaaaah"?)

Each day Jon Marius calls up a different doctor and asks them how long his "aaah" has to be. Find out if Jon Marius would waste his time going to the given doctor.

Input

The input consists of two lines. The first line is the "aaah" Jon Marius is able to say that day. The second line is the "aah" the doctor wants to hear. Only lowercase 'a' and 'h' will be used in the input, and each line will contain between 0 and 999 'a's, inclusive, followed by a single 'h'.

Output

Output "go" if Jon Marius can go to that doctor, and output "no" otherwise.

Sample Input 1	Sample Output 1
aaah aaaaah	no

Sample Input 2	Sample Output 2
aaah ah	go

This page is intentionally left (almost) blank.

Problem B
Bread Sorting
Problem ID: bread

Photo by Jim Champion

Michael works in a bakery. At the end of his shift, his boss wants the breads sorted in a certain order. She can't seem to decide on which order, though – every day there seems to be a new one – to Michael's despair. Michael has worked there for a while now and has learned a neat trick with his wooden bakery paddle. He can take three breads next to each other on his paddle and throw them up in the air such that when they land, the right-most has moved to the left-most position, and the two other breads have moved one place to the right. In other words, he can rotate to the right a subsequence of breads of length three.

Before the end of the shift, his coworkers place the breads in a long line. Michael would like to sort the line of breads using his paddle trick. He can take any three consecutive breads along the line on his paddle, rotate them, and then put them back. Sometimes though, it does not matter how many times he uses his paddle – the line of bread just doesn't seem to be possible to sort the way the boss wants...

Input

The first line of input contains a positive integer n, $3 \leq n \leq 100\,000$. Then two lines follow: On the first line, a permutation of the integers 1 through n describing the order in which the breads are lined up. On the second line, a permutation of the integers 1 through n describing how Michael's boss wants the breads sorted.

Output

Output "`Possible`" if Michael can sort the breads with his paddle in the order prescribed by his boss, otherwise "`Impossible`".

Sample Input 1

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

Sample Output 1

```
Possible
```

Sample Input 2

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

Sample Output 2

```
Impossible
```

This page is intentionally left (almost) blank.

Problem C
Cookie Selection
Problem ID: cookieselection

Photo by Andrew Magill.

As chief programmer at a cookie production plant you have many responsibilities, one of them being that the cookies produced and packaged at the plant adhere to the very demanding quality standards of the Nordic Cookie Packaging Consortium (NCPC).

At any given time, your production line is producing new cookies which are stored in a holding area, awaiting packaging. From time to time there are also requests from the packaging unit to send a cookie from the holding area to be packaged. Naturally, you have programmed the system so that there are never any packaging requests sent if the holding area is empty. What complicates the matter is the fact that representatives of the NCPC might make a surprise inspection to check that your cookies are up to standard. Such an inspection consists of the NCPC representatives demanding that the next few cookies sent to the packaging unit instead be handed over to them; if they are convinced that these cookies look (and taste) the same, you pass the inspection, otherwise you fail.

Fortunately, the production plant has invested in a new measurement thingamajig capable of measuring cookie diameters with a precision of 1 nanometre (nm). Since you do not have the time to always be on the lookout for cookie craving NCPC inspectors, you have decided that a sensible strategy to cope with the inspections is to *always* send a cookie having the median diameter among all cookies currently in the holding area to the packaging unit on a request. If there is no cookie exhibiting the median diameter in the holding area, you instead send the smallest cookie larger than the median to the packaging unit, hoping it will please the NCPC inspectors. This means that, if the cookies were sorted in order of ascending diameter, and if there was an odd number c of cookies in the holding area, you would send the cookie at position $(c+1)/2$ in the sorted sequence, while if there was an even number c of cookies in the holding area, you would send the cookie at position $(c/2)+1$ in the sorted sequence to the packaging unit on a request.

Input

Each line of the input contains either a positive integer d, indicating that a freshly baked cookie with diameter d nm has arrived at the holding area, or the symbol '#', indicating a request from the packaging unit to send a cookie for packaging. There are at most 600 000 lines of input, and you may assume that the holding area is empty when the first cookie in the input arrives to the holding area. Furthermore, you have read somewhere that the cookie oven at the plant cannot produce cookies that have a diameter larger than 30 centimetres (cm) (or 300 000 000 nm).

Output

A sequence of lines, each indicating the diameter in nm of a cookie sent for packaging, in the same order as they are sent.

Sample Input 1	Sample Output 1
1 2 3 4 # # # #	3 2 4 1

Sample Input 2

```
1
#
2
#
3
#
4
#
```

Sample Output 2

```
1
2
3
4
```

NCPC 2012

Problem D
Doorman
Problem ID: doorman

Clip art from Microsoft.

The doorman Bruno at the popular night club Heaven is having a hard time fulfilling his duties. He was told by the owner that when the club is full, the number of women and men let into the club should be roughly the same. When the night club opens, people wanting to enter the club are already lined up in a queue, and Bruno can only let them in one-by-one. He lets them in more-or-less in the order they are lined up. He can however decide to let the second person in the queue cut the line and slip into the club before the person in front. This will no doubt upset the person first in line, especially when this happens multiple times, but Bruno is quite a big guy and is capable of handling troublemakers. Unfortunately though, he is not that strong on mental calculations under these circumstances. He finds keeping track of the difference of the number of women and number of men let into the club a challenging task. As soon as the absolute difference gets too big, he looses track of his counting and must declare to the party people remaining in the queue that the club is full.

Input

The first line of input contains a positive integer $X < 100$ describing the largest absolute difference between the number of women and number of men let into the club, that Bruno can handle. The second line contains a string consisting solely of the characters 'W' and 'M' of length at most 100, describing the genders of the people in the queue, in order. The leftmost character of the string is the gender of the person first in line.

Output

The maximum number of people Bruno can let into the club without loosing track of his counting. You may assume that the club is large enough to hold all the people in the queue.

Sample Input 1	Sample Output 1
1 MWWMWMMWM	9

Sample Input 2	Sample Output 2
2 WMMMMWWMMMWWMW	8

This page is intentionally left (almost) blank.

Problem E
Eco-driving
Problem ID: ecodriving

Photo by Bil Simser

My colleague Elisabeth is lazy, both at work and when going to work. She never wants to do more than necessary, which also applies to her journey to work. Her goal is to use a minimum amount of energy, which she achieves by braking and accelerating as little as possible. This method applies to all wheel-based transport means she owns.

Elisabeth has earlier optimized her route by trial and error, but now wants your help to find the optimal path. She has provided a map with N junctions and R straight one-way roads between the junctions. If a road is dual way, it is represented as two one way roads.

Fortunately, Elisabeth often works night shifts, so braking and acceleration are only needed when turning in a junction, as there is no other traffic on the roads. Her goal is to find a route where the maximum turning angle in the junctions is minimized, as this means maximized speed. However, the route can not be too long.

Input

The first line of input contains three space separated integers J, R, D ($2 \leq J \leq 200, 1 \leq R \leq 39\,800, 1 \leq D \leq 1\,000\,000$). J is the number of junctions, R is the number of one way roads which are connecting two junctions, and D is the maximum distance, in meters, that Elisabeth wants to travel. The road network is such that there is no path, that Elisabeth may want to use, which has the length L such that $D < L < D * (1 + 1e - 6)$.
Then follow J lines with two integers X and Y, ($-100\,000 \leq X, Y \leq 100\,000$), the distinct coordinates in meters on a flat ground. Elisabeth lives at junction 1 and her work is at junction J.
After that, R lines follow with two integers A and B. This should be interpreted as the one way road between the one-indexed source and destination junctions A and B ($1 \leq A, B \leq J$).

Output

Output one line with the maximum turning angle of the route that thas the maximum turning angle as low as possible. The turning angle must be output in degrees and have an absolute or relative error at most 10^{-6}. If there is no such route that is short enough, output "`Impossible`" on a single line.

Sample Input 1	Sample Output 1

5 6 500
-100 0
-100 100
0 200
100 100
100 0
1 2
1 3
2 3
3 4
3 5
4 5
	90.00000000

Sample Input 2

```
5 6 450
-100 0
-100 100
0 200
100 100
100 0
1 2
1 3
2 3
3 4
3 5
4 5
```

Sample Output 2

```
126.86989765
```

Sample Input 3

```
5 12 440
-100 0
-100 100
0 200
100 100
100 0
1 2
1 3
3 1
2 3
3 2
3 4
4 3
3 5
5 3
4 5
5 4
5 1
```

Sample Output 3

```
Impossible
```

Problem F
Food Review
Problem ID: foodreview

Photo by Nelson Pavlosky

Frida is a writer for Cosmopolitan who writes restaurant reviews. She enjoys it a lot, but it seems that, throughout the years, she has reviewed all the restaurants on Earth. It's now time to move one level up; she is going to review the food served by the airlines, so that the readers can make better decisions on which flights to take.

Her boss gave her a list of flight connections that she needs to review for the upcoming issue of Cosmopolitan. She knows that they serve the same food in both directions of every flight, so she only needs to take it once. She realized that she will need to take some additional flights, because she can not make all reviews using only flights in the list from her boss. Therefore she did some quick research and made a list of additional flights which she might take. She will not review the food on these flights; they will only be used so that she can make all the reviews.

Frida's goal is to make all the reviews while spending the least money on flight tickets. Her office is in Stockholm, so she starts and ends her journey there. Each flight is both ways between two cities and has a fixed price in both directions. You can assume that it is possible to finish all the reviews using some of the additional flights.

For the purposes of this problem we ignore the price Frida has to pay for accommodation and we also ignore the departure and arrival times of flights by assuming that every flight is very often and reasonably short. We only focus on the total price of the flights.

Input

The first line contains 2 space separated integers $N, R, (2 \leq N \leq 13, 0 \leq R \leq 78)$, where N is the number of airports mentioned in the input and R is the number of flights to review. The airports are numbered $1, \ldots, N$ and Stockholm has number 1.

The next R lines describe the R flights to review. Each line contains 3 space separated integers $a, b, c, (1 \leq a, b \leq N, 1 \leq c \leq 10\,000)$, where a, b denote 2 distinct airports and c is the cost of the flight in Swedish kronor in both directions. No pair of 2 cities is listed twice.

The next line contains an integer F, $(0 \leq F \leq 200)$, the number of additional flights available. The next F lines contain descriptions of flights in the same format as above and there may be more flights between a pair of cities. You may assume that it is possible to make all the reviews using some of these additional flights.

Output

Output one line with one integer – the lowest total cost of flight tickets, such that Frida can make all the reviews and return back to Stockholm.

Sample Input 1	Sample Output 1

5 3
1 2 1000
2 3 1000
4 5 500
2
1 4 300
3 5 300
	3100

Sample Input 2

```
6 5
1 2 1000
2 3 1000
1 3 1000
2 4 1000
5 6 500
2
2 5 300
4 6 300
```

Sample Output 2

```
5100
```

Problem G
Galactic Warlords
Problem ID: galactic

Correct version

Phto by NASA, ESA, G. Illingworth, D. Magee, and P. Oesch (University of California, Santa Cruz), R. Bouwens (Leiden University), and the HUDF09 Team.

Will the galaxy see peace at last? All the warlords have gathered to divide all of space between themselves. The negotiations have come quite far and the warlords have finally agreed on a peaceful way of deciding who gets what. The 2-dimensional galactic map must first be divided into sectors by splitting it along a set of infinite lines. The warlord with the largest battle fleet will choose one sector, then the warlord with the second largest fleet will choose some other sector and so on, until everyone has gotten a sector. This is then repeated until there are no sectors left.

Different sets of lines have been suggested, and it is up to you to present these alternatives to the meeting. To make sure that there will be peace, you are ready to modify the suggestions slightly. You have some experience with warlords and know that no warlord will settle for less space than anyone else, so for there to be peace, all of them must get the exact same area on the map. Since space is infinite, so is the map. Some sectors will therefore have infinite area, so that is the amount of space everyone will want. How many extra lines will you have to add to make sure each warlord can get at least one sector with infinite area?

Input

The first line of input contains two positive integers W and N, $(1 \leq W, N \leq 100)$ denoting the number of warlords and the number of lines in the suggested division of space. This is followed by N lines each containing four integers x_1, y_1, x_2 and y_2, each with an absolute value no higher than 10 000. This means that one line is intersecting the two points (x_1, y_1) and (x_2, y_2) on the galactic map. These two points will not be the same.

Output

Output the number of lines you will have to add to this suggestion to satisfy all warlords.

Sample Input 1

```
2 1
1 1 -2 0
```

Sample Output 1

```
0
```

Sample Input 2

```
5 3
0 5 5 5
0 0 1 1
2 2 3 3
```

Sample Output 2

```
1
```

This page is intentionally left (almost) blank.

Problem H
Horror List
Problem ID: horror

Photo by Janet Hudson

It was time for the 7th *Nordic Cinema Popcorn Convention*, and this year the manager Ian had a brilliant idea. In addition to the traditional film program, there would be a surprise room where a small group of people could stream a random movie from a large collection, while enjoying popcorn and martinis.

However, it turned out that some people were extremely disappointed, because they got to see movies like *Ghosts of Mars*, which instead caused them to tear out their hair in despair and horror.

To avoid this problem for the next convention, Ian has come up with a solution, but he needs your help to implement it. When the group enters the surprise room, they will type in a list of movies in a computer. This is the so-called *horror list*, which consists of bad movies that no one in the group would ever like to see. Of course, this list varies from group to group.

You also have access to the database *Awesome Comparison of Movies* which tells you which movies are directly similar to which. You can assume that movies that are similar to bad movies will be almost as bad. More specificly, we define the *Horror index* as follows:

$$HI = \begin{cases} 0 & \text{if movie is on horror list. This overrides the other definitions.} \\ Q+1 & \text{if the worst directly similar movie has HI = Q} \\ +\infty & \text{if not similar at all to a horrible movie} \end{cases}$$

Input

The first line of input contains three positive integers N, H, L ($1 \leq H < N \leq 1000, 0 \leq L \leq 10000$), where N is the number of movies (represented by IDs, ranging from 0 to $N-1$), H is the number of movies on the horror list and L is the number of similarities in the database.

The second line contains H unique space-separated integers x_i ($0 \leq x_i < N$) denoting the ID of the movies on the horror list.

The following L lines contains two space-separated integers a_i, b_i ($0 \leq a_i < b_i < N$), denoting that movie with ID a_i is similar to movie with ID b_i (and vice verca).

Output

Output the ID of the best movie in the collection (highest Horror Index). In case of a tie, output the movie with the lowest ID.

Sample Input 1	Sample Output 1

6 3 5
0 5 2
0 1
1 2
4 5
3 5
0 2
	1

Sample Input 2

```
6 2 3
5 2
0 5
0 1
3 4
```

Sample Output 2

```
3
```

Problem I
Infiltration
Problem ID: infiltration

Illustration by: Mike Holmesfour, used under CC Attribution Noncommercial 3.0 Unported License

Rookie Pirate Captain Jack Albatross has secretly laid anchor in a hidden bay not far from Port Wine, the local meeting point for grizzled pirates and denizens of the sea. Disguising as a messenger, you have infiltrated the service of the dreaded legendary Captain Stevie, and you are entrusted with carrying small encrypted messages between Captain Stevie and his staff of shipmates. Your task is to break the encryption of the messages in aid of young Captain Albatross, giving him a much needed edge in his quest for peg-legged fame. Fortunately, you (in contrast to Captain Albatross) have been around for a while and you've come to know what words to expect in written pirate messages. The following is a table of words having common occurrences:

"be"	"our"	"rum"	"will"
"dead"	"hook"	"ship"	"blood"
"sable"	"avenge"	"parrot"	"captain"

In a (mixed alphabet) substitution cipher, each letter of the plain text is replaced with another, in this case arbitrary but unique, letter from the alphabet. Given a text encrypted with a substitution cipher and the given set of the twelve known words above, your task is to decode the encrypted text and provide us and Captain Albatross with the plain text. This can be accomplished if and only if there is a unique substitution mapping from the encrypted letters to plain text letters such that

1. a subset S of the twelve words show up in the plain text, and
2. the number of different letters in the words in S is equal to the number of different letters in the encrypted text.

Note that not all the known words may be present in the plain text and that replacements are not mutual ('a' being encrypted by 'h' does NOT necessarily mean that 'h' is encrypted by 'a').

Input

A text encrypted by a substitution cipher.

The encrypted text is given on one line containing at most 200 characters from the set ['a'-'z',' ']. Only the letters ['a'-'z'] of the plain text are encrypted, the spaces are kept unencrypted.

Output

The decrypted plain text if possible, or the string "`Impossible`" if the text cannot be uniquely decrypted using the set of known words.

Sample Input 1	Sample Output 1
`ex eoii jpxbmx cvz uxju sjzzcn jzz`	`we will avenge our dead parrot arr`

Sample Input 2	Sample Output 2
`wl jkd`	`Impossible`

Sample Input 3	Sample Output 3
`dyd jkl cs`	`Impossible`

This page is intentionally left (almost) blank.

Problem J
Juice
Problem ID: juice

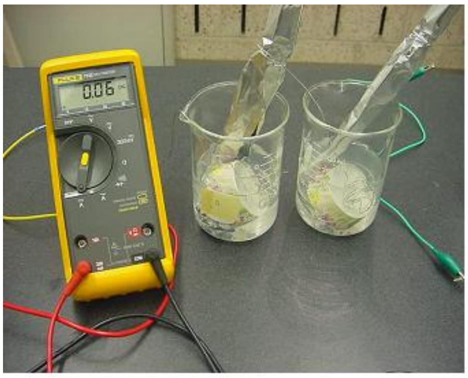

Photo by Joe Friedrichsen

In the Favela of Rio de Janeiro there was a flicker of light. After months of careful building, they finally connected the generator to the thousands of extension cords and the slum was illuminated by millions of bright lights…

However, the capacity of the extension cords was not enough to meet the energy demands of all the houses in the slum. Thus, the engineers had to carefully select which houses should be powered, and which should not, prior to connecting the power generator. Their idea was to power as many houses as possible, based on the energy demands of each house, and the capacity of the extension cords.

More specifically, the generator and each of the houses are represented by nodes, and extension cords are represented by edges between these. Thus, each node gets power from exactly one other node. In addition, each node except the generator node has a non-negative power demand. The generator produces an amount of energy that far surpasses the total capacity of the extension cords connected to it, and can thus be treated as an infinite energy source.

Given the same data, find out how many houses the engineers were able to cover the energy demands of.

Input

The first line of the input contains a single integer $0 \leq n \leq 1000$, the number of houses in the slum.
Then follow n lines in the format p_i r_i c_i, where $0 \leq p_i \leq n$ is the parent node of house i, $0 \leq r_i \leq 100$ is the energy demand of house i and $1 \leq c_i \leq 100$ is the capacity of the extension cord connecting house i to house p_i. The power generator has index 0.

Output

Output the maximum number of power requirements that can be met.

Sample Input 1	Sample Output 1

3
0 3 2
0 100 100
1 1 1
	2

This page is intentionally left (almost) blank.

Problem K
Kindergarten
Problem ID: kindergarten

Photo by Lars Plougmann

Every year the three friendly teachers at the kindergarten let their classes' kids change classes to form three new ones. Some kids, of course, are old enough to leave, but those who stay for another year are rearranged among the three teachers.

The teachers even let the kids have their say in the process. As friendship both comes and goes hastily in young years, each kid X ranks every other kid Y according to how glad X would be to have Y in her new class. In fact, X produces a preference list giving a total order of the other kids, i.e. there are no such things as ties – kids she would be equally glad to have as class mates.

The three teachers do not mind if the new classes formed are unbalanced in size, since they fill up their classes with new kids about to start their first year at the kindergarten. They do, however, want all the kids in their own new class to be different from last year, since even a teacher needs a break after a whole year with the same kids. They decide that a best partition into three classes under these premises is one where no kid is in the same class as a kid not listed among the top T entries on their preference list, for T as small as possible. Note that the kids in a new class may very well be the same as in an old one, but then with a new teacher!

Input

The first line of input contains a positive integer $n \leq 200$ giving the number of kids to be rearranged at the kindergarden. The kids are numbered 1 through n.
Then follow n lines describing the kids. The i-th row first contains the identifier of their current class' teacher (an integer 0, 1, or 2), and next the $n-1$ integers $\{1, 2, 3, \ldots, i-1, i+1, \ldots, n\}$ in some order, describing the class mate preference list of the i-th kid, in descending order.

Output

The smallest non-negative integer T, such that there is a partitioning of the kids in three new classes such that

- No kid has the same teacher as in their current class, and
- all kids' class mates are among the top T places of their preference lists, respectively.

Sample Input 1	Sample Output 1

6
0 2 3 4 5 6
0 1 3 4 5 6
1 6 5 4 2 1
2 6 5 3 2 1
1 1 2 3 4 6
2 1 2 3 4 5
	4

Sample Input 2	Sample Output 2

3
0 2 3
1 1 3
2 1 2
	0

This page is intentionally left (almost) blank.

Problem L
Luggage
Problem ID: luggage

Photo by Bernhard J. Scheuvens

Johan has a really boring job at the airport. It is his duty to make sure no bags collide when sliding onto the circular conveyor belt in the luggage pick-up zone. All pieces of luggage are loaded onto a straight conveyer belt which ends above the circular conveyor belt. Johan is then manually varying the speed of the straight conveyor so that no collisions occur. He would rather set a constant speed so that he can go to the fika room. There is a sensor at the start of the long conveyor belt so that he can get the positions of all pieces of luggage into his computer. Now all he needs is a program giving him the maximum speed that will avoid all collisions.

The circular conveyor belt moves at a constant speed of 1 m/s. If a piece of luggage is dropped on the circular belt within one meter of the position of an other piece of luggage they may collide, so this should be avoided. Some bags might be picked up on the first round, but you can not know in advance which will be left. The straight conveyor can be set to any speed between 0.1 m/s and 10 m/s.

Input

The first line of input contains two positive integers N and L ($1 \leq N \leq L \leq 1000$), where N is the number of pieces of luggage and L is the length of the circular conveyor belt in meters. The second line contains N unique space-separated numbers x_i ($0 \leq x_i \leq 1000$) with two digits after the decimal point, denoting luggage positions in meters.

Output

Output the maximum speed v in m/s ($0.1 \leq v \leq 10$) that makes sure no collisions will occur, or "`no fika`" if there is no such speed. The answer is considered correct if it has an absolute error of at most 10^{-9}.

You may assume that when an optimal v exists, each speed in the interval $[v - 10^{-9}, v]$ will also be a valid speed.

Sample Input 1	Sample Output 1
2 3 0.00 2.00	2

Sample Input 2	Sample Output 2
3 4 0.05 1.00 3.50	0.5

This page is intentionally left (almost) blank.