

Olympic Average

Some sports, such as gymnastics, use a "*trimmed mean*," in which the highest and lowest scores are discarded and the remaining ones averaged. This is also known as the Olympic average, due to its use in the Olympic events, to make the score robust to a single outlier judge.

You are a game developer. Your game Quarter Life 3 is now available on Sbeam. You have released multiple updates and patches for this game to add new features, improve balance, fix bugs, etc. Since it has been very popular among game players, you received lots of reviews on Sbeam each time you release a new patch. Your boss asked you to evaluate the total feedback for each update. And you decided to use *trimmed mean* to calculate the average review score after each update.

Each time you release an update, you will receive some new reviews. Then you will find two reviews on Sbeam, one with the largest score and one with the smallest score. You will remove those two reviews from Sbeam as they are outliers. Finally you can calculate the *trimmed mean* from the remaining reviews. All the remaining reviews will be kept for next update.

Calculating mean is easy. Your task is to calculate the difference between the removed scores for each update.

Input

The first line contains an integer n , $1 \leq n \leq 5000$, the number of released patches.

The subsequent n lines contains a sequence of non-negative numbers separated by whitespaces. The numbers in the $(i+1)$ -st line give the reviews scores for the i -th update.

The first number in each of these lines, k , $0 \leq k \leq 100,000$, is the number of reviews and the subsequent k numbers are positive integers of the review scores.

The review scores on Sbeam are integers ranged from 1 to 1,000,000, inclusive. The total number of all review scores is no bigger than 1,000,000.

And there will be at least 2 reviews on Sbeam after receiving reviews for each update.

Output

Print n lines, the i -th line contains the difference between the largest score and the smallest score for the i -th update.

Example 1

Input :

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

Output :

```
1
1
```

Example 2

Input :

```
5
2 1 1
3 2 3 1
4 5 1 5 10
0
1 2
```

Output :

```
0
```

2
9
3
3

3D Labyrinth

You are in a 3D labyrinth of Conflict Empire and you want to escape it. The labyrinth could be represented as a three dimensional array where every cell is either empty or filled by rock. In every minute, you can travel in any direction parallel to the edges of the labyrinth, i.e. north, south, east, west, up and down, by one cell. You are not allowed to step onto a rock cell nor allowed to travel out of the labyrinth without reaching the exit point.

Given the starting point and the exit point, your task is to determine the minimum time needed to escape this labyrinth.

Input

The first line of the input contains three integers K, N and M ($1 \leq K, N, M \leq 30$), indicating the number of layers, rows and columns of the labyrinth, respectively. Then there follows K blocks of N lines each containing M characters. Each character describes one cell of the labyrinth, where **#** indicates a cell of rock and **.** indicates an empty cell. The starting position is indicated by **S** and the exit by **E**. Note: there will be a single blank line after each level.

Output

Print one line **Escaped in x minute(s)**. where **x** is the minimum time needed to escape this labyrinth, or **Trapped!** if it is impossible to escape.

Example 1

Input:

3 4 5

S....

.###.

.##..

###.#

#####

#####

##.##

##...

#####

#####

#.###

####E

Output:

Escaped in 11 minute(s).

Example 2

Input:

1 3 3

S##

#E#

###

Output:

Trapped!

Pocket Money

In Light Kingdom, there are \$5, \$10, \$20, \$50 and \$100 notes and 5c, 10c, 20c, 50c, \$1 and \$2 coins. Kou's parent is going to give Kou some pocket money and your task is to figure out in how many ways that amount may be made up. For example, 20c could be made up in 4 ways: $4 \times 5c$, $10c + 2 \times 5c$, $2 \times 10c$ and $1 \times 20c$.

Input

The input consists of a single line. This line contains one number M ($0.00 < M \leq 300.00$, with two decimal places), representing the amount of money Kou will receive.

Output

You should print one line consisting of M (with two decimal places and right justified in a field of width 6), followed by the number of ways in which M may be made up (right justified in a field of width 17).

Example 1

Input :

0.20

Output :

0.20

4

Example 2

Input :

2.00

Output :

2.00

293

Max Product Subarray

In computer science, the maximum product subarray problem is the task of finding a contiguous subarray with the largest product, within a given one-dimensional array $A[1..n]$ of numbers.

Formally, the task is to find indices i and j with $1 \leq i \leq j \leq n$, such that the product

$$a[i] * a[i+1] * \dots * a[j-1] * a[j]$$

is as large as possible.

For example, for the array of values $[5, -2, -10, -1]$, the contiguous subarray with the largest product is $[5, -2, -10]$, with product 100.

Write a program to calculate such product for the given input array.

Input

The first line contains 1 integer n , the number of elements in the input array, $1 \leq n \leq 100$.

In the next line, there are n integers indicating the elements of the array, $-10^5 \leq A[i] \leq 10^5$ for $1 \leq i \leq n$.

Output

Output one integer followed by a newline, indicating the product of the maximum product subarray.

Example 1

```
Input:
4
5 -2 -10 -1

Output:
100
```

Example 2

```
Input:
1
-9

Output:
-9
```

Deficit Cycle

You drive a truck between N cities to earn your living. You live in city 0 and the highways among those cities have the following properties:

- Highways are one-way only.
- Every highway has two endpoints and they shall not be the same city.
- There is at most one highway in either direction between any pair of cities.
- Starting from your home (city 0), you may travel to any other city using those highways.
- You can earn some money by traveling through a highway (someone will pay you for helping transport goods from one city to another).
- The money earned could be negative (if the fuel is more expensive than what the trip pays).

A deficit cycle is a driving path starting from one city, going through several highways and getting back to the same city with the negative amount of total earnings for the trip. You may repeatedly loss money if you happen to drive along a deficit cycle again and again. You wonder if such a deficit cycle exists given the highway map of those N cities.

Input

The first line of the input contains two integers N ($1 \leq N \leq 1000$) and M ($0 \leq M \leq 2000$), indicating the number of cities and the number of highways. Each of the following M lines contains three integers u , v and w , indicating there is a highway by which you can travel from city u to city v and you can earn w dollars. ($0 \leq u, v < N$, $u \neq v$, $-1000 \leq w \leq 1000$)

Output

Print one line containing **possible** or **not possible** to indicate if there exists a deficit cycle.

Example 1

```
Input:
3 3
0 1 1000
1 2 15
2 1 -42
```

```
Output:
possible
```

Example 2

```
Input:
4 4
0 1 10
1 2 20
2 3 30
3 0 -60
```

```
Output:
not possible
```