Virus

Due to the outbreak of unknown virus in 2029 in the mountains of Timbet, all the monks at Senpou Temple are required to do physical checkups at Iosefka's Clinic. All checkups are scheduled on the same day. Each monk gets instructions in which he is given

1. his unique patient number from the set { 1 … n }
2. the time of the day when he is supposed to show up at Iosefka's Clinic
3. a list of doctors' offices that he is to visit in the specified order

Doctors' offices in Iosefka's Clinic are numbered with numbers from the set { 1 … m }.

Since a doctor can only check one monk at a time, if several people show up to a doctor's office at time t, they form a queue in increasing order of their numbers and join the end of the queue already formed by monks who arrived earlier.

If at time t in front of office x there is a queue of people who arrived earlier or at time t, then the first monk from the queue enters offic x. This monk exits the office after one time unit and at time t+1 appears at the next office from his list of offices to visit. At that time the next person from the queue enters office x.

If a monk was supposed to show up at the clinic at time t, then at time t he shows up at the first doctors' office on his list. If a visit at office x at time t was for the given monk the last visit on his list, then at time t+1 this monk leaves the clinic.

Your task is to find the time when the last monk leaves the Iosefka's Clinic.

Input

The first line of input contains 2 natural numbers n and m, 1 <= n, m <= 10000, giving the number of the monks and the number of doctors' offices. Each of the following n lines contains a sequence of natural numbers. Among these lines, line i (1 <= i <= n) has the following format

```
 t k g1 g2 ... gk
```

meaning that the ith monk arrives at time t and has to visit k offices in the order given by g1 g2 … gk where each gj is a number of doctor's office, 1 <= gj <= m. We have that 0 <= t <= 1,000,000 and there is no more than 1,000,000 visits scheduled for a day at the clinic.

Output

Print one line giving the time when the last monk leaves the hospital.

Example 1

```
  Input:
  5 3
  1 3 3 2 1
  0 7 2 3 1 1 1 1 2
  2 1 1
  1 2 3 3
  4 3 1 1 1

  Output:
  12
```

Example 2

```
  Input:
  5 10
  3 1 6
  2 3 3 2 8
  2 1 4
```

```
2 4 7 9 9 6
0 2 8 7
```

Output:
6

Grade Sort

You are given the midterm exam scores for all students in all of NYU campuses, 0 ≤ score ≤ 100.

To report the midterm grades, you need to display all the scores in ascending order.

Input

The first line contains one integer N. N is the total number of grades, 1 ≤ N ≤ 10,000,000.

In the next line, there are N integers indicating the grades.

Output

Print a line with N space separated integers. These integers are the grades of the midterm sorted in ascending order. The output ends with a newline.

Warning: This problem may have potentially very large inputs/outputs. Use fast IO operations, please.

Example 1

```
Input:
4
50 60 100 90

Output:
50 60 90 100
```

Example 1

```
Input:
6
99 92 93 92 93 91

Output:
91 92 92 93 93 99
```

Ayu's Candies

Ayu has three flavors of candies: banana flavored, grape flavored and cherry flavored. She also has three boxes numbered 1, 2 and 3, each of which contains some of those candies. Now Ayu wants to move candies such that after the movement, each box contains the candies of only one flavor and no two boxes contain same flavored candies. Your task is to help Ayu find the minimum number of candy movements.

Input

The input consists of only one line, containing 9 integers. The first three integers represent the number of banana flavored, grape flavored and cherry flavored candies in the first box, the second three integers represent the number of banana flavored, grape flavored and cherry flavored candies in the second box, and the last three integers represent the number of banana flavored, grape flavored and cherry flavored candies in the third box. It is guaranteed that the total number of candies never exceeds 2^{31}.

For example, the input `10 15 20 30 12 8 15 8 31` indicates that there are 10 banana flavored candies in box 1, 12 grape flavored candies in box 2 and 31 cherry flavored candies in box 3.

Output

Print one line `S x` indicating which flavor of candies go in which box to minimize the number of candy movements. `S` is a string of three letters `G`, `B` and `C` (representing the flavor of grape, banana and cherry, respectively) indicating the flavor associated with each box. `x` is the minimum number of candy movements.

If there is more than one solution, print the alphabetically smallest string as `S`.

Example 1

```
Input:
1 2 3 4 5 6 7 8 9

Output:
BCG 30
```

Example 2

```
Input:
5 10 5 20 10 5 10 20 10

Output:
CBG 50
```

Sprinklers

Farmer John has a large field, and he is thinking of planting sweet corn in some part of it. After surveying his field, FJ found that it forms a horizontal strip l meters long and w meters wide.

There are n sprinklers installed at the horizontal center line of the strip. For each sprinkler you are given its radius of operation and its position as the distance from the left end of the center line. Farmer John wants to plant sweet corn in the whole field, but he wishes to turn on as few sprinklers as possible to save the water.

Find the minimum number of sprinklers that need to be turned on in order to water the entire field.

Input

The first line contains integer numbers n, l and w with $1 \le n \le 10{,}000$, $1 \le l \le 10{,}000{,}000$, and $1 \le w \le 100$. The next n lines contain two integers giving the position x ($0 \le x \le l$) and radius of operation r ($1 \le r \le 1000$) of a sprinkler.

The picture above illustrates the first case from the sample input.
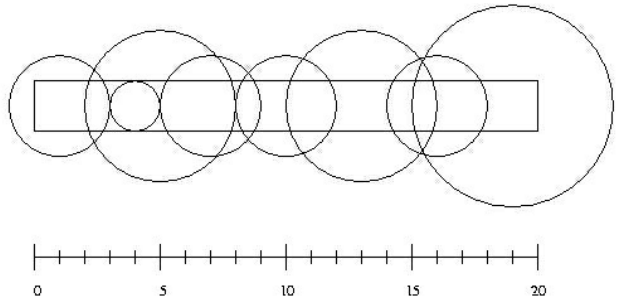
Output

Output one integer followed by a newline: the minimum number of sprinklers needed to water the entire strip.

If it is impossible to water the entire strip, output -1.

Example 1 (shown in the image above)

```
Input:
8 20 2
5 3
4 1
1 2
7 2
10 2
13 3
16 2
19 4

Output:
6
```

Example 2

```
Input:
3 10 1
3 5
9 3
6 1

Output:
2
```

Example 3

```
Input:
3 10 1
5 3
1 1
9 1

Output:
-1
```

Find Sums

You are given a multiset of N integers (multiset means that the repeated values are allowed) and a target value S. Your task is to find all distinct subsets of the given multiset for which the sum of all the elements in the subset is equal to S.

Input

The first line of the input contains two integers S (1 <= S <= 1,000) and N (1 <= N <= 12), indicating the target sum and the number of values in the multiset, respectively.

The second line contains N integers, all of which are between 1 and 100 - these are the elements of the multiset.

Output

First, print a line `Sums of S:` where S is the value given in the input. Then print one line for every subset satisfying the condition or a line containing `NONE` if there is no such subset.

For every subset, numbers are printed in decreasing order and separated by a plus sign (`+`). The subsets themselves are sorted lexicographically in decreasing order, i.e. they are sorted by their first integer, then the second integer in case of tie, and so on. Additionally, the subsets you print should not contain repetitions (i.e., you should never print two lines that are identical).

Example 1

```
Input:
4 6
4 3 2 2 1 1

Output:
Sums of 4:
4
3+1
2+2
2+1+1
```

Example 2

```
Input:
5 3
2 1 1

Output:
Sums of 5:
NONE
```

Example 3

```
Input:
400 12
50 50 50 50 50 50 25 25 25 25 25 25

Output:
Sums of 400:
50+50+50+50+50+50+25+25+25+25
50+50+50+50+50+25+25+25+25+25+25
```