Dancing Hippos

NYU is preparing a new attraction for the graduation ceremonies: A Dancing Hippos Act. The new president of atractions was able to locate a supplier who can provide a large number of dancing hippos. The idea for the act is to have them dancing on a platform that will be raised above the graduates. The hope is to have as many hippos dancing up on the elevated platform as possible. But the platform has limited weight capcity and we would not want to have new graduates squashed by the falling hippos.

Given the weight capacity `W` of the platforms, the number of dancing hippos available `N`, and the weights of all the hippos, `w_1, w_2, ..., w_N`, we need to figure out the largest number of dancing hippos that can be placed on the platform without endangering the graduates.

Input

First two lines contains `0<=W<=10^8` and `0<=N<=10^5` as described above. The net line contains `N` weights of the dancing hippos, `0<=w_i<=10^8`.

Output

Maximum number of hippos that can be safely placed on the platform.

Example 1

```
Input
50
4
76 56 64 69


Output
0
```

Example 2

```
Input
150
7
45 32 78 56 22 63 37

Output
4
```

Example 3

```
Input
10000
10
87 21 15 43 60 61 71 57 79 20

Output
10
```

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 <= i <= j <= n, such that the product

a[i] * a[i+1] * ... * 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 <= n <= 100.

In the next line, there are n integers indicating the elements of the array, -10^5 <= A[i] <= 10^5 for 1 <= i <= 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
```

Array Partition

Given an array A of N natural numbers and a number M (1 <= M <= N). Split the given array into M consecutive subarrays such that the maximum sum of the values in the subarrays is minimal.

For example: A = [3,4,2,1], M = 3. The optimal split is {3}, {2}, {2,1}. The maximum sum of all subarrays is 4, which is minimum possible for 3 splits. (Any other split would result in at least one subarray whose sum of values is greater than 4.)

Input

The first line contains 2 integer N, M as described above. 1 <= M <= N <= 10^5.

In the next line, there are N natural numbers indicating the elements of the array. 0 <= A[i] <= 10^9 for 1 <= i <= N

Output

Output one integer followed by a newline, indicating the minimal maximum sum of any subarray in the optimal split.

Example 1

```
Input:
4 3
4 3 2 1

Output:
4
```

Example 2

```
Input:
4 1
4 3 2 1

Output:
10
```

Example 3

```
Input:
3 2
1 100 2

Output:
101
```

Ayu's Shopping

Ayu is keen on desserts. She found that she will run out of them soon so she decided to buy some more today. Her favorite dessert shop offers several flavors (banana, grape, vanilla, etc.) for every kind of dessert (candy, cake, ice-cream, etc.). Ayu has M dollars and she wants to spend as much of that money as possible. She also doesn't want to get too fat so she decided to buy only one flavor of each kind of dessert. The question is how much money will she spend.

Input

The first line of the input contains two integers M (1 <= M <= 200) and N (1 <= N <= 20), indicating the amount of money Ayu has and the number of different kinds of desserts. Each of the following N lines contains K + 1 integers where K is the first integer of that line (1 <= K <= 20), indicating the number of flavors for this kind of dessert. The following K integers indicate the price for each flavor of the dessert (these are integers).

Output

Print one line consisting of one integer indicating the maximum amount of money spent to buy one flavor of each kind of dessert, or `no solution` if there is no solution.

Example 1

```
Input:
100 4
3 8 6 4
2 5 10
4 1 3 3 7
4 50 14 23 8

Output:
75
```

by picking 8, 10, 7 and 50.

Example 2

```
Input:
20 3
3 4 6 8
2 5 10
4 1 3 5 5

Output:
19
```

Example 3

```
Input:
5 3
3 6 4 8
2 10 6
4 7 3 1 7

Output:
no solution
```