Load Balance

You are an operator of a super computing center and in control of M nodes. One day, a research institute from Light Kingdom submitted N computational tasks. Given the computational power needed for each task, you are to distribute the tasks among the available nodes. Restriction: every node can process up to two tasks. You also want to distribute the workload as evenly as possible, i.e. minimize the following imbalance value

$$Imbalance = \sum_{i=1}^{M} (Average - Load_i)$$

where $Average$ is average load per node and $Load_i$ is the load of the i-th node.

Input

The first line of the input contains two integers M (1 <= M <= 5) and N (1 <= N <= 2M), indicating the number of nodes you control and the number of tasks, respectively.

The second line contains N integers, each of which represents the computational power required for a task. Numbers on these line are between 1 and 1000.

Output

Print one line `IMBALANCE = I` where `I` is the minimum imbalance value rounded to 5 decimal places followed by a new line character.

Example 1

```
Input:
2 3
6 3 8

Output:
IMBALANCE = 1.00000
```

Example 2

```
Input:
3 5
51 19 27 14 33

Output:
IMBALANCE = 6.00000
```

Example 3

```
Input:
5 9
1 2 3 5 7 11 13 17 19

Output:
IMBALANCE = 11.60000
```

Finding Routes

Ethan Hunt wants your help to get through a high security building for a top secret mission. The building has some cameras installed at several places which are constantly monitored by the security.

Ethan has a map of the building. Each corner is identified by a positive integer less than 21. Initially Ethan is at position 1.

Write a program to find the possible routes that Ethan can take to reach the nearest corner to the target location of the building.

Input

First line consists of an integer n which is the number of corners closest to the target. n<21

Next several lines consists of pairs of positive integers separated by a space which are the adjacent corners of the building which does not have any cameras in the way (for instance, if pair 4 7 is on the line then the path between corner 4 and 7 has no cameras on the path. Also there are no other corners on that path).

The last line consists of a pair of 0's

Output

The output must list each possible route on a separate line with each corner written in the order in which they appear on the path. Also, it should include a line stating the number of paths possible for reaching the desired location in the form "There are n routes from the initial position to corner x.", where n is the number of routes and x is the desired target corner. Only include the routes which do not pass through any corner more than once (this is to avoid moving unnecessarily around in circles as time is crucial).

Example 1

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

Output:
1 2 3 4 6
1 2 3 5 6
1 2 4 3 5 6
1 2 4 6
1 3 2 4 6
1 3 4 6
1 3 5 6
There are 7 routes from the initial position to corner 6.
```

Example 2

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

```
2 5
5 7
3 1
1 8
4 6
6 9
0 0
```

```
Output:
1 3 2 5 7 8 9 6 4
1 3 4
1 5 2 3 4
1 5 7 8 9 6 4
1 6 4
1 6 9 8 7 5 2 3 4
1 8 7 5 2 3 4
1 8 9 6 4
There are 8 routes from the initial position to corner 4.
```

Grocery Delivery

Nia hates shopping and she really despises grocery shopping. She orders her groceries from a local store. In preparation for drone deliveries, the store has new rules about packing items for delivery: - they will deliver exactly two boxes to the customer (even if the order is small and one of the boxes is empty) - the capacity of each box is the same and equal to W.

Figure 1:

Nia knows the volume occupied by each item on her shopping list: the ithitem has the volume vi. With the new rules, she has to prioritize the items that she orders since not everything can fit in the two boxes. Her list is arranged in the order of priority: from the most important item to the least important one (i.e., the ith item is always more important than the i+1st item). As she prepares the list to send to the store, she only orders the highest priority items and as soon as she comes to an item that is too large to fit in the remaining space in the boxes, she skips that item and all the items below.

The question is how many items from her list she will be able to order and which item should be placed in which box.

Input

The first line contains 2 integers N and W, indicating the number of items and the capacity of a box, 1 <= N <= 200, 1 <= W <= 10000.

The second line contains N integers, indicating the volume of items in Nia's list. They are in the order of priority as described above.

Output

The first line of output should specify the number of items that can be ordered.

For each item that can be ordered, output a line containing "1st" if the item is to be placed in the 1st box and "2nd" if the item is to be placed in the 2nd box. If several arrangements of the items are possible, any one will do.

Example 1

```
Input:
5 10
1 2 100 3 4

Output:
2
1st
1st
```

Explanation of example 1:

Nia has to remove the 3rd item since it's too large (this means that the 4th and the 5th items are also removed from the order, even though they could fit in the second box). The first 2 items will be ordered. Any arrangement of the items in boxes will do.

Example 2

```
Input:
7 50
25 30 10 10 15 7 8

Output:
6
1st
2nd
2nd
2nd
1st
1st
```

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}, {4}, {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
```

Giant Squad

Conflict Empire is fighting a war against Light Kingdom. The commander of Conflict Army is building a squad of giants. Giants are huge creatures, typically a dozen feet tall, inhabiting the territory of Conflict Empire. Giants are so tall that their height differences are also very large, preventing orders from being effectively transmitted. To address this issue, the commander decides to form squads with only odd number of giants with different heights and select the captain in such a way so that the number of giants in the squad who are shorter than that captain is equal to the number of giants who are taller than that captain. You are given heights of all giants in that squad and you are asked to determine the height of the captain.

Input

The input consists of a single line, starting with an integer N ($1 < N < 11$, N is odd), the number of giants in that squad, followed by N integers representing heights of the giants of the squad. Those N integers will be between 11 and 20 (inclusive). Additionally, heights will be given in strictly increasing or decreasing order.

Output

You should print a single integer x, where x is the height of the captain.

Example 1

```
Input:
5 19 17 16 14 12

Output:
16
```

Example 2

```
Input:
5 12 14 16 17 18

Output:
16
```