"A" Секретный код

Time limit	2 seconds
Memory limit	256 megabytes

Now Marty is in the past, and he wants to get back home to his 1985. He has already made his parents fall in love with each other, and has found plutonium. What is left to do is to activate the time machine and start the time travel. However, there is a problem for Marty. To activate the time machine one needs to enter the secret code. Only Doc know the secret code. All that Marty knows is that it consists of several different characters. Also he knows the length of the code. While Marty is waiting for Doc, he is trying to guess the code and enters different character combinations.

Now Doc has returned and entered the code. Marty would like to know how close he was to entering the correct code. For each Marty's attempt to enter the code, find the number of correct characters in the code that are at their correct positions, and the number of correct characters that are however at incorrect positions.

Input format

The first line contains s — the correct secret code. The code consists of uppercase letters of the English alphabet and digits. All characters in the code are different.

The second line contains a positive integer n ($1 \le n \le 10^5$) — the number of Marty's attempts.

Each of the following n lines contain

information about Marty's attempts. Each of these lines contains the code of the same length as *s* that contains uppercase letters of the English alphabet and digits. All characters within one code are different.

Output format

For each Marty's attempt print two integers: *a* and *b* — the number of correct characters at correct positions, and the number of correct characters, but at incorrect positions.

Examples

Input data

```
BACKTO1985
3
BACKTO1958
BACKON1985
TOYEAR1985
```

Output data

```
8 2
8 1
```

"B" Chaos

Time limit	2 seconds
Memory limit	256 megabytes

Setting up dominoes and watching them fall toppling each other has become too boring for Doctor Brown. So he has invented the new, more mathematical, time waste.

The board contains n integers. Doc repeatedly performs the following moves:

- He chooses three numbers a, b and c written at the board and erases them.
- He chooses two numbers from the triple a, b and c and calculates their average value, rounding it down if necessary, getting the number d.
- He writes the number d to the board two times.

For example, if the board contains 1, 2 and 4, after erasing them from the board, Doc can write two 1-s (average of 1 and 2 rounded down), two 2-s (average of 1 and 4 rounded down) or two 3-s (average of 2 and 4) on the board. The process is over when there are two numbers left on the board. Clearly these two numbers are the same.

Once Marty was watching Doc's actions, and it really looked as some random sequence of moves. But it was not, Doc claims that he has made his moves in such way, that the numbers left on the board are maximal possible. Marty doesn't completely trust Doc. He has remembered the initial numbers on the board, and wants you to find the maximal possible value of the two numbers left on the board after performing the sequence of moves.

Input format

The first line contains one integer n ($3 \le n \le 10^5$) — the number of integers written on the board.

The second line contains n integers a_i ($1 \le a_i \le 10^9$) — the numbers on the board.

Output format

Output one integer — the maximal possible value of the two numbers left on the board in the end of Doc's actions.

Examples

Input data

```
3
1 4 2
```

Output data

```
3
```

Input data

```
5
3 3 3 3 3
```

Output data

3

"C" New Adventure of Marty and Doc

Time limit	2 seconds
Memory limit	256 megabytes

After Doc's return from 1885 using time train, he is now excited to try other means of transportation as time machines, so he has created the time airplane! But during the first test of the time airplane, its engines failed, and the airplane crashed. Doc has ejected himself from the cabin and successfully landed using safety parachute, but the airplane has fallen down to a large field and shattered into small parts.

Now the parts of the crashed airplane need to be evacuated and recycled. Marty has drawn the plan of the field, it is a rectangle with n rows and m columns. Each cell of the rectangle contains zero or more parts of the airplane. Marty has decided to put recycling plant at one cell of the field, and bring all parts of the airplane to that cell. A special Doc's robot will bring all parts of the airplane to the recycling plant. The robot can perform the following three actions:

- Move from its current cell to a neighboring cell that shares a side with it.
- If there is an airplane part in the current cell, pick it up. The robot can carry only one part at a time, so if he is already carrying a part, he cannot pick up another one.
- If the robot is in the same cell as the recycling plant and is carrying an airplane part, recycle it.

The robot will start from the cell where the recycling plant is located.

Now Marty needs to find out what cell should be used to build the recycling plant to minimize the number of robot's actions to recycle all the airplane parts. Help him!

Input format

The first line contains integers n and m ($1 \le n \cdot m \le 10^6$) — the size of the field.
The i-th of the following lines contains the description of the i-row of the field: m integers$a_{i,j}$ ($0 \le a_{i,j} \le 10^6$) — the number of airplane parts in the corresponding field cell.

Output format

Print three integers: r, c and x ($1 \le r \le n$, $1 \le c \le m$) — row and column of the optimal position of the recycling plant and the total number of actions that the robot needs to recycle all the airplane parts. If there are several optimal positions for the recycling plant, print any of them.

Examples

Input data

```
3 3
0 0 0
0 1 0
0 0 0
```

Output data

2 2 2

"D" Teams Creation

Time limit	2 seconds
Memory limit	256 megabytes

Dr Emmett Brown has decided to change his job and is now working as a Computer Science teacher in a high school. The Dr Brown's class has n students. Dr Brown wants to run a programming contest for his students. But his classroom only has k computers, so he needs to run a team contest.

Dr thinks that the teamwork would be good for the students if the students in the team all have close levels of their skills. For each student Emmett Brown knows its skill level a_i. He has decided to create teams in such way that for any two teams there is a number x such that students in one team have skill level at most x, and in the other team all students have skill level at least x. There must be exactly k teams, each team must have at least one student, but there is no upper limit for the number of students in one team.

Help Doctor to find out how many ways are there for him to create the teams. Two ways are different if there are two students that are in the same team in of the them, and in different teams in the other. You should output the number of ways modulo $10^9 + 7$.

Input format

The first line contains two integers n and k ($1 \le n, k \le 2\,000$) — the number of students in the class and the number of teams that must

be created.

The second line contains n integers a_i ($1 \le a_i \le n$) — the skill levels of all students.

Output format

Output one integer — the number of ways to create the teams modulo $10^9 + 7$.

Examples

Input data

```
3 2
1 2 3
```

Output data

```
2
```

Input data

```
7 3
2 4 3 4 3 3 2
```

Output data

```
53
```

"E" Maximal sum

Time limit	2 seconds
Memory limit	256 megabytes

Marty wants to get back to the future from the past. But the computer system of his time machine is broken, so he needs to make some calculations by himself and then enter the results.

Marty has two arrays of integers: $a[1..n]$ and $b[1..m]$. For each b_j Marty needs to find the segment $a[l..r]$ such that each element in it is greater or equal to b_j and the sum of elements $a[l] + a[l+1] + ... + a[r]$ is maximal possible. These sums must be entered into time machine's computer system to get Marty back to the future.
Help him, write the program to solve this problem.

Input format

The first line of input contains two integers n and m ($1 \le n, m \le 10^5$) — the sizes of arrays $a$and b, respectively.

The second line contains n integers a_i ($-10^9 \le a_i \le 10^9$).

The third line contains n integers b_j ($-10^9 \le b_j \le 10^9$).

Output format

Output m integers, the j-th of them must be the required maximal sum for b_j. If there is no such segment in a array, output 0 instead.

Examples

Input data

```
5 5
-1 2 3 4 -5
-5 4 10 2 -1
```

Output data

```
9 4 0 9 9
```

Input data

```
5 5
3 -2 3 -5 -3
-1 -2 -3 -4 -5
```

Output data

```
3 4 4 4 4
```