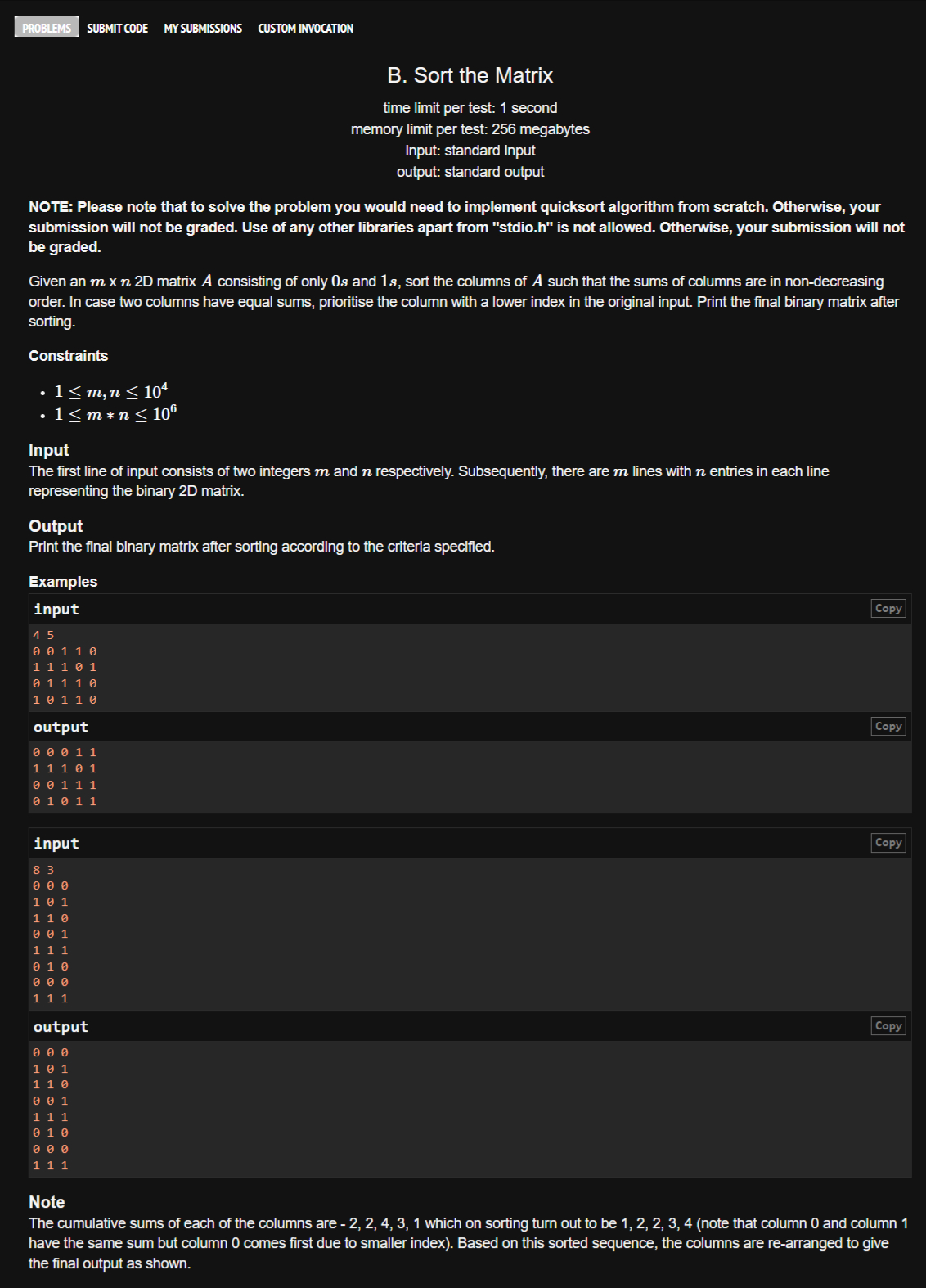

B. Sort the Matrix

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

NOTE: Please note that to solve the problem you would need to implement quicksort algorithm from scratch. Otherwise, your submission will not be graded. Use of any other libraries apart from "stdio.h" is not allowed. Otherwise, your submission will not be graded.

Given an m x n 2D matrix A consisting of only $0s$ and $1s$, sort the columns of A such that the sums of columns are in non-decreasing order. In case two columns have equal sums, prioritise the column with a lower index in the original input. Print the final binary matrix after sorting.

Constraints

- $1 \leq m, n \leq 10^4$
- $1 \leq m * n \leq 10^6$

Input

The first line of input consists of two integers m and n respectively. Subsequently, there are m lines with n entries in each line representing the binary 2D matrix.

Output

Print the final binary matrix after sorting according to the criteria specified.

Examples

input

```
4 5
0 0 1 1 0
1 1 1 0 1
0 1 1 1 0
1 0 1 1 0
```

output

```
0 0 0 1 1
1 1 1 0 1
0 0 1 1 1
0 1 0 1 1
```

input

```
8 3
0 0 0
1 0 1
1 1 0
0 0 1
1 1 1
0 1 0
0 0 0
1 1 1
```

output

```
0 0 0
1 0 1
1 1 0
0 0 1
1 1 1
0 1 0
0 0 0
1 1 1
```

Note

The cumulative sums of each of the columns are - 2, 2, 4, 3, 1 which on sorting turn out to be 1, 2, 2, 3, 4 (note that column 0 and column 1 have the same sum but column 0 comes first due to smaller index). Based on this sorted sequence, the columns are re-arranged to give the final output as shown.