United International University

Data Structure and Algorithms II Laboratory

Spring 241 CSE 2218

Sec: F Marks: 25

Assignment 02

Copying from online/friend or submitting someone's work as your own is a violation of ethical standards and may lead to serious consequences (Including negative marking)

1. Minimum Swaps for Bracket Balancing 10

You are given a string of 2N characters consisting of N '[' brackets and N ']' brackets. A string is considered balanced if it can be represented in the form S2[S1] where S1 and S2 are balanced strings. We can make an unbalanced string balanced by swapping adjacent characters. Calculate the minimum number of swaps necessary to make a string balanced.

Input String	Minimum Swaps	Explanation
[ ]] [ ] [	2	In the input string "[]][][", the first swap involves positions 3 and 4, resulting in "[][]][". The second swap involves positions 5 and 6, making the final string "[][][]". Therefore, a minimum of 2 swaps is needed to make the string balanced.
[[ ] [ ]]	0	The input string "[[][]]" is already balanced, so no swaps are needed. The minimum number of swaps is 0.

2. Problem 02 15

In the country of XYZ, there is an annual programming competition where participants are given a set of coding problems to solve. This year, the organizers have introduced a new rule to make the competition more challenging. There are n participants, numbered from 1 to n. Participant n is considered the top coder, and the organizers want to ensure that this participant doesn't win any of the problems.

Each participant submits their solutions to m different problems, numbered from 1 to m. You know the scores each participant received for each problem. The only action allowed to prevent the top coder from winning any problem is to disqualify the results of some problems. The top coder will win a problem if the sum of their

scores on all non-disqualified problems is strictly greater than the analogous sum for every other participant.

Your task is to disqualify the minimum number of problems to ensure that the top coder doesn't win any problem. Note that a solution always exists, as disqualifying all problems will result in all participants having a score of 0, and the top coder won't win any problem.

Input:
The first line of the input contains two integers n and m ($2 \le n \le 100$; $1 \le m \le 100$) — the number of participants and the number of problems. The next m lines contain the scores for each participant on each problem, with n numbers on each line. In the i-th line, the j-th number is $a_{i,j}$ — the score of participant j on problem i ($0 \le a_{i,j} \le 1000$).

Output:
In the first line, output the integer k — the minimal number of problems that need to be disqualified. In the second line, output k integers — the indices of disqualified problems, in any order. If there are multiple ways to disqualify results for k problems, output any one of them.

Input	Output
4 3 5 3 6 8 4 7 2 5 2 5 4 9 3 6 7 8	1 3
3 4 6 3 8 4 7 5 2 5 7 3 2 6	2 1 3