CSE 4304-Data Structures Lab. Winter 2022
Batch: CSE 21
Date: August 17, 2023,
Target Group: All
Topic: Strings, sorting, binary search

<u>**Instructions**</u>:
- Regardless you finish the tasks in the lab, you have to submit the solutions in the Google Classroom. In case I forget to upload the tasks there, CR should contact me. The deadline will always be at 11.59 PM of the day in which the lab has taken place.
- Task naming format: fullID_T01L02_2A.c/cpp
- If you find any issues in the problem description/test cases, comment in the google classroom.
- If you find any test case that is tricky that I didn't include but others might forget to handle, please comment! I'll be happy to add.
- Use appropriate comments in your code. This will help you to easily recall the solution in the future.
- Obtained marks will vary based on the efficiency of the solution.
- Do not use <bits/stdc++.h> library.
- Modified sections will be marked with BLUE color.

Group	Tasks
2B	
2A	
1A	1 6 7 8
1B	1 6 7 8

Task-01:

Alexander wants to fight for the Coding Club. In each round, there will be **N** soldiers with various powers. There will be **R** rounds to fight and in each round, his power will change. With power **P**, he can kill all the soldiers whose power is less than or equal. The sum of powers of the soldiers will be considered as the total point for that round. After each round, all the soldiers who are dead in the previous round will be reborn. So in each round, there will be **N** soldiers to fight.

Your job is to count the number of soldiers that he can kill in each round.

Input will start with **N** & **R** followed by **N** power levels of the soldiers. The next lines will contain the **R** power levels of Alexander for the different rounds.

Input	Output
7 3 1 2 3 4 5 6 7 3 10 2	3 7 2
10 8 5 12 13 17 25 35 41 42 43 55 2 39 13 22 73 29 0 35	0 6 3 4 10 5 0 6

Note:
-   Use Binary Search.
-   Do not use any built-in function.

Task 06:

Two arrays are given, one containing the positions and another containing the values. Your job is to rearrange the values based on the positions.

The first line contains the position values. Keep taking input in the first line until -1 is given. The following line will contain the values corresponding to those positions.

Input	Output
3 1 2 -1 32 54.7 -2	54.7 -2 32
5 1 6 4 3 2 -1 3.75 3.88 3.75 3.92 3.99 3.84	3.88 3.84 3.99 3.92 3.75 3.75

Note: **05 Bonus marks for O(N) solution.**

Task 7:

Given your total obtained marks in the final exam (out of 100), your task is to return the obtained grade.

The first line of input will contain two numbers 'level' and 'trials'. Then each line will contain a number and the associated grade level.
After the levels are inserted, the trails will be inserted. Each trial will contain an integer value.
For each trial, you have to print the corresponding grade.

Input	Output
10 5 39 F 40 D 45 C 50 C+ 55 B- 60 B 65 B+ 70 A- 75 A 80 A+ 13 43 67 89 61	F D B+ A+ B
6 7 39 F 50 C 60 B 70 A- 80 A+ 90 A* 0 100 89 77 60 61 59	F A* A+ A- B B C
7 9 39 F 40 D 50 C 60 B 70 A- 80 A+ 90 A* 39 35	F F F A* A+ A- B B C

0 100 89 77 60 61 59	

Note: Try to find the most efficient solution. Utilize binary Search.

Task 08:

One measure of 'undersortedness' in a sequence is the number of pairs or entries that are out of order with respect to each other. For instance, in the letter sequence 'DAABEC', the measure is 5, since D is greater than four letters to its right and E is greater than one letter to its right. This measure is called the number of inversions in the sequence. The sequence 'AACEDGG' has only one inversion (E and D) and it is nearly sorted. On the other hand, the sequence 'ZWQM' has 6 inversions; it is as unsorted as can be, exactly in the reverse order.

You are responsible for cataloging a sequence of DNA strings (sequences containing only the four letters A, C, G, and T). However, you want to catalog them not in alphabetical order, but in the order of 'sortedness'- from the 'most sorted' to 'least sorted'. All the strings are of the same length.

The first line of input will contain two numbers 'length' and 'n' representing the length of each sequence and the total number of strings.

Your task is to arrange the string from 'most sorted' to 'least sorted' order. On the right of each string, show the degree of unsortedness.

Input	Output
10 6 AACATGAAGG TTTTGGCCAA TTTGGCCAAA GATCAGATTT CCCGGGGGGA ATCGATGCAT	CCCGGGGGGA 9 AACATGAAGG 10 GATCAGATTT 11 ATCGATGCAT 17 TTTTGGCCAA 36 TTTGGCCAAA 37