National Institute of Technology Calicut
Department of Computer Science and Engineering
Third Semester B. Tech.(CSE)
CS2092D Programming Laboratory
Assignment 2

Submission deadline (on or before):

- 11:00 AM, Sunday, 13.08.2023

Policies for Submission and Evaluation:

- You must submit your assignment in the Eduserver course page, on or before the submission deadline.
- Ensure that your programs will compile and execute without errors in the Linux platform.
- During the evaluation, failure to execute programs without compilation errors may lead to zero marks for that evaluation.
- Detection of ANY malpractice related to the lab course can lead to awarding an F grade in the course.
- **Any queries or clarifications on the assignment questions should be submitted on or before 11:00 AM, Saturday, 12/08/2023 in the below spreadsheet**: https://docs.google.com/spreadsheets/d/1
hUNTcRPhTQFHcpKBfM/edit?usp=sharing

  Any queries after that will not be entertained.

Naming Conventions for Submission

- Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar, .tar, .gz). The name of this file must be

  `ASSG<NUMBER>_<ROLLNO>_<FIRST-NAME>.zip`

  (Example: *ASSG2_BxxyyyyCS_LAXMAN.zip*). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive.

- The source codes must be named as

  `ASSG<NUMBER>_<ROLLNO>_<FIRST-NAME>_<PROGRAM-NUMBER>.c`

  (For example: *ASSG2_BxxyyyyCS_LAXMAN_1.c*). If you do not conform to the above naming conventions, your submission might not be recognized by our automated tools, and hence will lead to a score of 0 marks for the submission. So, make sure that you follow the naming conventions.

Standard of Conduct

- Violation of academic integrity will be severely penalized. Each student is expected to adhere to high standards of ethical conduct, especially those related to cheating and plagiarism. Any submitted work MUST BE an individual effort. Any academic dishonesty will result in zero marks in the corresponding exam or evaluation and will be reported to the department council for record keeping and for permission to assign F grade in the course. The department policy on academic integrity can be found at: https://minerva.nitc.ac.in/cse/sites/default/files/attachments/news/Academic-Integrity.pdf.

General Instructions

- Programs should be written in C language and compiled using C compiler in Linux platform. **Submit the solutions to questions 1 to 4 through the submission link in Eduserver**.

Note

- **In the binary search implementation, if the no of elements in the array are even, always take the left element as the middle element out of the two potential candidates for becoming the middle element.**

QUESTIONS

1. Imagine you are a software developer working on a project that involves optimizing search algorithms for large datasets. Your team is tackling a specific problem of finding missing elements in ascending sequences. You receive a task to write a program for the following problem:

   Given an ascending sequence, denoted by an array 'A', its size 'n' and an integer 'k', the objective is to determine the k-th missing contiguous element (k-th element in the sequence of missing elements) in this sequence. If the k-th missing element is not there for the sequence, the program should output -1. Additionally, you are asked to print the total number of array cells probed(accessed) by your program during the search process to prove that you have written the most efficient code for the problem.

   Hint: (You can use Binary Search for minimizing the no of probes taken by your program on an average).

   Input Format:

   - The first line is an integer $n \in [1, 10^6]$.
   - The second line contains 'n' integer numbers within the range $\in [1, 10^5]$ separated by a space , representing the array elements.
   - The third line of the input is an integer $k \in [1, 10^9]$

   Output Format:

   - A single line contains two integer values separated by a space, representing k-th missing number and the number of array probes done by the program.

   Sample Input 1:
   6
   1 2 3 6 8 10
   3
   Sample Output 1:
   7 3

   Sample Input 2:
   5
   4 9 10 14 19
   5
   Sample Output 2:
   6 3

2. Anup is training for a marathon that will take place in N days. He needs to run M practice sessions to prepare for the marathon, and the i-th session requires Time[i] minutes of running. Anup has decided on some rules for the session:-

   - He will only move to the next session after completing the ongoing session.
   - Session once started will be completed on the same day.
   - He will sequentially complete the sessions.

Anup wants to distribute his training sessions over N days to minimize the maximum time he runs in a day. Consider Anup's day is 100^{100} minutes. Print the minimum value of the maximum amount of time he needs to allocate each day to complete all sessions in N days.

Input Format:

- First line contains the number of days $N \in [1, 10^4]$ and the number of sessions $M \in [1, 10^4]$ separated by a space.
- The second line contains integer numbers within the range $\in [1, 10^9]$ separated by a space, representing minutes required by each session.

Output Format:

- Single integer representing the minimum value of the maximum amount of time he needs to allocate each day.

Sample Input 1:
4 7
2 2 3 3 4 4 1
Sample Output 1:
6

Sample Input 2:
3 5
1 2 2 3 1
Sample Output 2:
4

3. You are given an array of 'n' integers (both positive and negative) so that the sequence strictly increases until it reaches a peak, and then strictly decreases. This is known as a mountain array. You are given such an array. The peak element is an element at position 'i' such that:

   arr[i-1] < arr[i] and arr[i+1] < arr[i]

   You are asked to use the best known algorithm to minimize the number of comparisons done by your program on an average to find the peak (Hint: Binary Search).

   Along with the peak element, you should also print the number of comparisons done by your program for finding the peak.

   Input Format:

   - First line contains the number of elements 'n' in the array $n \in [1, 10^3]$.
   - Second line contains the elements in the array $arr \in [-10^3, 10^3]$.

   Output Format:

   - an integer containing the peak element.
   - an integer indicating the number of comparisons done by the program

   Sample Input 1:
   9
   1 3 5 6 7 8 5 3 2

   Sample Output 1:
   8
   6

   Sample Input 2:
   5

1 2 3 4 5

Sample Output 2:
5
5

4. In a distant land, an ancient scroll contains a sequence of distinct numbers. The scroll's magical power is said to be activated only when the sequence is rotated at some point. The sequence represents the wisdom of the ancients and contains various hidden insights. However, accessing this powerful knowledge is not an easy task. The sequence of numbers is sorted in ascending order and then rotated at an unknown point, making it difficult to navigate and find specific elements.

   You are a brave adventurer and have managed to obtain the scroll. As you unravel its secrets, you discover a cryptic message hinting at a "secret number" hidden within the rotated sequence. The message warns that this secret number will only be revealed to one who can uncover it with the minimum number of comparisons.

   Given a rotated sequence of distinct elements and the secret number you seek, your task is to write a program that finds the secret number's position within the sequence(starting with zero) by doing minimum number of comparisons (on an average). If the secret number is not present, your program should print -1.

   Along with the secret number's position, you also need to print the number of comparisons your program made to find the secret number's position.
   (Hint: Use binary search)

   Input Format:

   - The first line is an integer $n \in [1, 10^7]$, total numbers in the scroll.
   - The second line contains 'n' integer numbers within the range $\in [0, 10^8]$ separated by a space, indicating a sequence of numbers inside the scroll.
   - The third line is an integer $k \in [1, 10^7]$, the secret number.

   Output Format:

   - A single line contains two integer values separated by a space, representing the secret number's position and the number of comparisons done by the program.

   Sample Input 1:
   9
   5 6 7 8 9 10 1 2 3
   10
   Sample Output 1:
   5 3

   Sample Input 2:
   4
   3 5 1 2
   5
   Sample Output 2:
   1 1

Best Wishes