

**National Institute of Technology Calicut**  
**Department of Computer Science and Engineering**  
**Second Semester B. Tech.(CSE)**  
**CS1092E Program Design Laboratory**

**Assignment #1**

**Submission deadline (on or before): 23/01/2024, 08:00 PM [Questions 1- 5],  
30/01/2024, 08:00 PM [Questions 6, 7]**

**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: *ASSG6\_BxyyyyyCS\_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: *ASSG6\_BxyyyyyCS\_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: [http://cse.nitc.ac.in/sites/default/files/Academic-Integrity\\_new.pdf](http://cse.nitc.ac.in/sites/default/files/Academic-Integrity_new.pdf).

**General Instructions**

- Programs should be written in C language.
- Check your programs with sufficiently large values of inputs with in the range as specified in the question.
- Global and/or static variables should not be used in your program.

**QUESTIONS**

1. Write a program that takes an array of  $N$  integers, sorts it using Insertion Sort and prints the sorted array.

**Input Format:**

- The first line of your input reads the size  $N$  of the array of integers
- The second line takes elements of the array.
- $1 \leq N \leq 100000$ .
- Array elements  $x$  are in the range  $-100000 \leq x \leq 100000$ .

**Output Format:**

- The sorted array of integers in ascending order.

**Sample Input 1 :**

38 27 43 3 9 82 10

**Sample Output 1 :**

3 9 10 27 38 43 82

**Sample Input 2 :**

5 17 23 11 13

**Sample Output 2 :**

5 11 13 17 23

2. Imagine you are managing a library system, and the books in the library are currently stored in an array. To organize the books, you decide to implement the Insertion Sort algorithm. Implement a C program to sort the books based on their titles in lexicographical order.

**Note:** One book name can be of 100 characters containing special characters, integers, characters.

**Input Format:**

- The first line of your input reads the size  $N$  of the array of books in a library. Assume  $1 \leq N \leq 100000$
- The second line takes the name of  $N$  books.

**Output Format:**

- The first line of output gives sorted array of books in lexicographical order

**Sample Input 1 :**

5

Harry Potter, The Great Gatsby, To Kill a Mockingbird, 1984, Pride and Prejudice

**Sample Output 1 :**

1984, Harry Potter, Pride and Prejudice, The Great Gatsby, To Kill a Mockingbird

**Sample Input 2 :**

5

Programming in C, Algorithms in Java, Python Crash Course, Data Structures in C++, Web Development Basics

**Sample Output 2 :**

Algorithms in Java, Data Structures in C++, Programming in C, Python Crash Course, Web Development Basics

3. Consider an online store that sells electronic devices with each product having a name and a price(price ranging between 1000 to 100000 rupees). The store owner wants to display the products in ascending order based on their prices. The store has a limited number of products, so performance is not a significant concern. Write a 'C' program to sort the list of products by their prices using the best sorting mechanism for this case. [Hint: Selection sort is best for relatively small datasets. Make use of structures in C while implementing.]

**Input Format:**

- The first line of your input reads the size  $N$  of the array of product structures. Assume  $1 \leq N \leq 100000$ .
- Next inputs the name and price of  $N$  products. Assume that names are strings of size at most 100.

**Output Format:**

- A sorted array of structures in ascending order of their prices.

**Sample Input 1:**

```
3
AAAA
15000
BBBB
10000
CCCC
11500
```

**Sample Output1:**

```
BBBB  CCCC  AAAA
10000 11500 15000
```

**Sample Input 2:**

```
2
Arjun
20000
Akshay
15000
```

**Sample Output2:**

```
Akshay  Arjun
15000   20000
```

4. Given an array 'arr' of size ' $N$ ' of integers and an integer ' $K$ '. Find the ' $K$ 'th largest element in that array.[Hint: Use selection sort mechanism.]

**Input Format:**

- The first line contains integer ' $N$ '.
- Next line contains ' $N$ ' space-separated integers.
- The last line contains the value of ' $K$ '.
- $1 \leq N \leq 100000$
- $1 \leq arr[i] \leq 100000$
- $1 \leq K \leq N$

**Output Format:**

- The ' $K$ 'th largest integer from the array.

**Sample Input 1:**

```
5
2 6 3 9 1
4
```

**Sample Output 1**

```
6
```

**Sample Input 2:**

```
10
11 27 61 33 13 29 90 71 55 88
4
```

**Sample Output 2:**

```
29
```

5. Given an array 'arr' of size 'N' of integers. Your task is to sort the elements in the ascending order using bubble sort, print the sorted array and the number of swaps used in the bubble sort algorithm.

**Input Format:**

- The first line of your input contains 1 integer, it inputs the size  $N$  of the array.
- The second line inputs the elements of the array.
- $1 \leq N \leq 100000$
- $1 \leq arr[i] \leq 100000$

**Output Format:**

- In the first line, you print the sorted array.
- In the second line, you print the number of swaps.

**Sample Input 1:**

```
5
2 6 3 9 1
```

**Sample Output 1:**

```
1 2 3 6 9
5
```

**Sample Input 2:**

```
4
27 61 33 13
```

**Sample Output 2:**

```
13 27 33 61
4
```

6. Imagine a digital music library with thousands of songs, each identified by a unique ID number. These songs need to be sorted efficiently for quick retrieval and seamless playback. Your task is to write a C program that implements the Quick Sort algorithm to arrange these songs in ascending order based on their ID numbers. For this task, you'll use the first element of each subarray as the pivot element for partitioning.

**Input Format:**

- The first line of input contains an integer  $n$  ( $0 \leq n \leq 10^4$ ), representing the number of songs in the library.
- The second line lists the ID numbers of each song, separated by spaces. ID numbers are positive integers within the range  $[1, 10^6]$ .

**Output Format:**

- A single line of output containing the sorted list of song IDs, separated by spaces.

**Sample Input 1:**

5  
32145 12568 98745 54321 65432

**Sample Output 1:**

12568 32145 54321 65432 98745

**Sample Input 2:**

10  
1 5 2 4 8 9 3 7 6 10

**Sample Output 2:**

1 2 3 4 5 6 7 8 9 10

7. Given an array 'arr' of size 'N' of integers. Sort the array in ascending order using merge sort.

**Input Format:**

- The first line of your input contains 1 integer, it inputs the size  $N$  of the array.
- The second line inputs the elements of the array.
- $1 \leq N \leq 100000$
- $1 \leq arr[i] \leq 100000$

**Output Format:**

- The output is the sorted array.

**Sample Input 1:**

4  
2 1 4 3

**Sample Output 1:**

1 2 3 4

**Sample Input 2:**

6  
6 5 4 3 2 1

**Sample Output 2:**

1 2 3 4 5 6