

# National Institute of Technology Calicut

Department of Computer Science and Engineering

Course: EE3038D Data Structures and Algorithms

## Assignment 1

Submission Date: 25.02.2022

### Policies for Submission and Evaluation

You must submit your assignment in the Moodle (Eduserver) course page, on or before the submission deadline. Also, ensure that your programs in the assignment must compile and execute without errors in any server. Failure to execute programs in the assignment without compilation errors may lead to zero marks for that program. Your submission will also be tested for plagiarism, by automated tools. In case your code fails to pass the test, you will be straightaway awarded zero marks for this assignment.

### Naming Conventions for Submission

Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar or .tar.gz). The name of this file must be *ASSG1\_ < ROLLNO > \_ < FIRSTNAME > .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 > \_ < FIRSTNAME > \_ < PROGRAM – NO > . < extension >*. (For example: *ASSG1\_BxxxxxyEE\_ABHAY\_1.c*). If you do not conform to the above naming conventions, your submission might not be recognized by some automated tools, and hence will lead to a score of 0 for the submission. So, make sure that you follow the naming conventions.

### Question 1

The Sorting Problem can be formally stated in terms of the input/output relationship as follows:

Input: A sequence of  $n$  numbers  $a_1, a_2, \dots, a_n$

Output: A permutation  $a_1, a_2, \dots, a_n$  of the input sequence such that  $a_1 \leq a_2 \leq \dots \leq a_n$

Let  $A[1 \dots n]$  be an array of  $n$  distinct numbers.

If  $i < j$  and  $A[i] > A[j]$ , then the pair  $(i, j)$  is called an inversion of  $A$ .

Write a program that uses the Merge-Sort algorithm for sorting a given input sequence of integers present in an array  $A$ . Your program must print the number of inversions in the input array.

**Input Format:**

- The first line of the input contains an integer  $n \in [0, 10^5]$ , the size of the array A.
- The second line lists the  $n$  elements in A, as space-separated integers in the range  $[-10^3, 10^3]$ .

**Output Format:**

- The first line of the output contains the elements of A in sorted order, separated by space.
- The second line of the output contains the number of inversions in the array.

**Sample Input:**

```
5
1 5 6 4 20
```

**Sample Output:**

```
1 4 5 6 20
2
```

## Question 2

Write a program to implement a Stack S using an array A of size  $n$ . The stack must support the functions: isEmpty, Push, Pop and isFull. Modify the isFull and/or PUSH functions to support the following functionality: If S is full when the PUSH function is called, allocate a new array B of size  $2n$ , copy all the elements of A into B, make A point to the array B, deallocate the old array A, and finally perform the PUSH operation on the new array A.

**Input Format:**

- The first line of the input contains an integer  $n \in [0, 10^5]$ , the size of the array A.
- Upcoming lines contain a character from 'i', 'd', or 't' followed by zero or one integer. The integer, if given, is in the range  $[-10^6, 10^6]$ .
- Character 'i' is followed by an integer separated by space. In this operation, the integer is inserted to the top of S.
- Character 'd' is to delete and print the deleted element from S.
- Character 't' is to 'terminate' the program.

**Output Format:**

- The output (if any) of each command should be printed on a separate line.
- For option 'i', if A is full, then print 1.
- For option 'd', print the deleted element. If A is empty, then print 1.

**Sample Input :**

4

d

i 8

i 10

i 11

i 12

i 13

d

d

t

**Sample Output:**

-1

1

13

12