

National Institute of Technology Calicut
Department of Computer Science and Engineering
Fourth Semester B. Tech.(CSE)-Winter 2021-22
CS2094D Data Structures Laboratory
Assignment #0

Submission deadline (on or before):

- 02.01.2022, 11:00 PM

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 using gcc compiler.
- 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.

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: *ASSG1_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: *ASSG1_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: http://cse.nitc.ac.in/sites/default/files/Academic-Integrity_new.pdf.

QUESTIONS

1. 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
```

2. Write a program to implement a Queue Q using an array A of size n . The queue must support the functions: `isEmpty`, `ENQUEUE`, `DEQUEUE` and `isFull`.

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 ', ' e ', ' f ', 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 a space. In this operation, the integer is inserted to the *tail* of Q . If the Q is full, then print 1.
- Character ' d ' is to delete and print the first element of Q .

- Character ‘e’ is to check whether the Q is empty or not.
- Character ‘f’ is to check whether the Q is full or not.
- 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 ‘d’, print the deleted element. If Q is empty, then print 1.
- For option ‘e’, if Q is not empty, then print -1 . If Q is empty, then print 1.
- For option ‘f’, if Q is not full, then print -1 . If Q is full, then print 1.

Sample Input :

```
5
i 8
i 10
d
i 12
d
d
d
e
i 18
f
i 1
i 5
e
t
```

Sample Output:

```
8
10
12
1
1
-1
1
-1
```

3. Assume that you are given the head pointer of an unsorted singly linked list L that contains n nodes, for some unknown integer n . Note that n is not part of the input. Write a program that implements the following function:

kLast: Takes as input the head pointer of a singly linked list L and an integer k , such that $k \leq n$, where n is the length of L and returns the $(n - k + 1)^{th}$ node in the list.

(Hint: You should create the Singly linked list L with the elements that is read from the console.)

Input format:

- The input should be read from the console.
- The first line contains the elements of L which are separated by a space.
- Second line is the integer k .

Output format:

- If $k \leq n$, print the $(n - k + 1)^{th}$ node in the list. Otherwise print -1.

Sample Input:

12 35 50 59 60 73 90

3

Sample Output:

60

Sample Input:

12 35 50 59 60 73 90

10

Sample Output:

-1