National Institute of Technology Calicut
Department of Computer Science and Engineering
Third Semester B.Tech.(CSE)
CS2092D Programming Laboratory
Assignment #6 - Part 1

Submission deadline (on or before):

- 10.11.2021, 9:00 AM

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 here.

General Instructions

- Programs should be written in C language and compiled using gcc compiler. **Submit the solutions to the questions through the submission link in Eduserver**.
- 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 menu driven program to implement a Stack S using a singly linked list. Each node in the list has an integer attribute *data* and a pointer attribute *next*. Keep *S.top* point to the first node (node at the front end) of the list. Your program must contain the following functions: (in function prototypes, S denotes a Stack, k an integer value, and x a node in the list. All operations should run in *O(1)* time.)

   - Main() - repeatedly reads a character '*i*', '*d*', or '*e*' from the terminal and calls the appropriate function until character '*t*' is entered.
   - Create-Node(k) - creates a new node with *data* value k and returns a pointer to the created node. The attribute *next* of the new node should be set as NULL.
   - Push(S, x) - inserts x as the new top node of S.
   - Pop(S) - deletes the most recently inserted node from S.
   - Stack-Empty(S) - checks whether the Stack is empty or not.

   Note:- For every Push() operation, the node x is to be created by calling the Create-Node() function.

   Input format:

   - Each line contains a character from '*i*', '*d*', '*e*', 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 node with this integer as data is inserted to the front end of S.
   - Character '*d*' is to delete the most recently inserted node and the deleted node's data value is printed.
   - Character '*e*' is to check whether the Stack is empty 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 S is empty, then print -1.
   - For option '*e*' if S is not empty, then print 1. If S is empty, then print -1.

   Sample Input :

   ```
   i 15
   i 25
   d
   i 2
   i 22
   d
   d
   d
   e
   d
   e
   t
   ```

   Sample Output:

   ```
   25
   22
   2
   ```

```
15
-1
-1
-1
```

2. Write a menu driven program to implement a Queue Q using a singly linked list. Queue is to be declared as a *struct* with two pointer attributes *head* and *tail*, pointing to the first node and the last node of the list respectively. Each node in the list is an object with an attribute *data* and a pointer attribute, *next*. Your program must contain the following functions: (in function prototypes, Q denotes a Queue, k an integer (k need not be distinct) and x a node. All operations should run in *O(1)* time.).

   - Main() - repeatedly reads a character '*i*', '*d*', or '*e*' from the terminal and calls the appropriate function until character '*t*' is entered.
   - Create-Node(k) - creates a new node with data value k and returns a pointer to the new node. The *next* field should be set as NULL.
   - QueueEmpty(Q) - checks whether the Queue is empty or not.
   - Enqueue(Q, x) - inserts x to the tail of Q.
   - Dequeue(Q) - deletes the first node from Q (check QueueEmpty() inside this function).

   Note:- For every Enqueue operation, the node x is to be created by calling Create-Node() function.

   Input format:

   - Each line contains a character from '*i*', '*d*', '*e*' 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 node with this integer as data is inserted to the tail of Q.
   - Character '*d*' is to delete the first node and the deleted node's key is printed.
   - Character '*e*' is to check whether the Queue is empty 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*' delete the first node and print the deleted node's key. 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.

   Sample Input

```
i 10
i 4
i 14
d
d
d
e
i 21
e
t
```

Sample Output

```
10
4
14
-1
1
```