

DATA STRUCTURES

ASSIGNMENT - IV

(FOR THE SIXTH LAB SESSIONS)

Assignments to be completed during lab sessions

- ✓ 1. Implement a stack with an array. The maximum size of the stack is user input.
- ✓ 2. Implement a circular queue with an array. The maximum size of the queue is user input.
- ✓ 3. Write a function to convert an infix expression to postfix expression. A operand could be any element in the set $\{0, 1, 2, \dots, 9\} \cup \{a, b, c, \dots, z\} \cup \{A, B, C, \dots, Z\}$. A operator could be any element in the set $\{+, -, *, /, \wedge\}$. The expression may also have parenthesis.
4. Write a function to evaluate a postfix expression. An operand could be any element in the set $\{0, 1, 2, \dots, 9\}$. A operator could be any element in the set $\{+, -, *, /, \wedge\}$.
5. Implement a stack with a circular doubly linked list that contains information about the first (head) node only. Now, implement a queue with two instances of the stack.
6. Implement a queue with a circular singly linked list that contains information about the first (head) node and the last (tail) node. Now, implement a stack with two instances of the queue. Reverse a string using the stack.

Additional assignments

1. Consider the following two structures.

```
struct book_t {
    int    ukey;      /* A unique key assigned to the book. */
    char *issn;      /* A pointer to the ISSN number of the book. */
    char *name;      /* A pointer to the name of the book. */
};
struct student_t {
    int    roll;      /* The roll no of the student. */
    char *name;      /* A pointer to the name of the student. */
};
```

Implement a generic singly linked list so that you can store a pointer to either of `book_t` and `student_t` in any node of the list with the following four operations defined on the list: (i) insert an element at the i th location of the list, (ii) delete the element from the i th location of the list, (iv) get the element at the i th location of the list, (iv) find the length of the list, (v) a generic print function that prints each element of the list, and (vi) a generic sort function that can sort the list using any of the member variables of the above two structures. Now, Consider a library management system. Say, there are b different books (`book_t`) and s number students (`student_t`). Create a list of books and a list of students. Sort and print both the lists using `ukey` and `roll`, respectively.

2. Implement two stacks in an array.
3. Rewrite all the programs associated with linked lists both in Assignment-II and in Assignment-III, such that, the implementations contain information about the first (head) node and the last (tail) node.