Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
2	2	2	2	2	10	

Assignment No. 01

Date of Performance: ……………………..………………

AIM: Write a program to implement stack as an abstract data type using linked list and use this ADT for conversion of infix expression to postfix, prefix and evaluation of postfix/prefix expression.

OBJECTIVE:

1) To understand the concept of abstract data type.
2) How different data structures such as arrays and a stacks are represented as an ADT.

THEORY:

1) What is an abstract data type?

An Abstract Data type is defined as a mathematical model of the data objects that make up a data type as well as the functions that operate on these objects. There are no standard conventions for defining them. A broad division may be drawn between "imperative" and "functional" definition styles. In general terms, an abstract data type is a *specification* of the values and the operations that has two properties:

- It specifies everything you need to know in order to use the data type
- It makes absolutely no reference to the manner in which the data type will be implemented.

When we use abstract data types, our programs divide into two pieces:

- The Application: The part that uses the abstract data type.
- The Implementation: The part that implements the abstract data type.

1) What is stack? Explain stack operations with neat diagrams.

In computer science, a **stack** is a last in, first out (LIFO) abstract data type and data structure. A stack can have any abstract data type as an element, but is characterized by only two fundamental operations: push and pop. The push operation adds an item to the top of the stack, hiding any items already on the stack, or initializing the stack if it is empty. A pop either reveals previously concealed items, or results in an empty stack. A stack is a restricted data structure, because only a small number of operations are performed on it. The nature of the pop and push operations also mean that stack elements have a natural order. Elements are removed from the stack in the reverse order to the order of their addition: therefore, the lower elements are those that have been on the stack the longest. A collection of items in which only the most recently added item may be removed. The latest added item is at the top. Basic operations are push and pop. Often top and isEmpty are available, too. Also known as "last-in, first-out" or LIFO.

Operations

An abstract data type (ADT) consists of a data structure and a set of **primitive**

- **Push** adds a new element
- **Pop** removes an element

Additional primitives can be defined:

- **IsEmpty** reports whether the stack is empty
- **IsFull** reports whether the stack is full
- **Initialize** creates/initializes the stack
- **Destroy** deletes the contents of the stack (may be implemented by re-initializing the stack)

3) Explain how stack can be implemented as an ADT.

User can Add, Delete, Search, and Replace the elements from the stack. It also checks for Overflow/Underflow and returns user friendly errors. You can use this stack implementation to perform many useful functions. In graphical mode, this C program displays a startup message and a nice graphic to welcome the user.

The program performs the following functions and operations:

- **Push**: Pushes an element to the stack. It takes an integer element as argument. If the stack is full then error is returned.
- **Pop**: Pop an element from the stack. If the stack is empty, then error is returned. The element is deleted from the top of the stack.
- **DisplayTop:** Returns the top element on the stack without deleting. If the stack is empty, then error is returned.

4) With an example explain how an infix expression can be converted to prefix and postfix form with the stack. (*Ans to be written by students*)

5) Write algorithm of

1. **Abstract Data Type Stack:**
2. **Infix to Prefix Conversion:**
3. **Infix to Postfix Conversion:**
4. **Postfix Expression Evaluation:**

INPUT:

Test Case	O/P
If Stack Empty	Display message Stack Empty
Stack Empty	Display message Stack Full

OUTCOME: After successful implementation of this assignment, we understood the concept of abstract data type (ADT). We also understood that how different data structures such as arrays and stacks are represented as an ADT.

FAQ:

1. What is data structure?
2. What are the different types of data structure?
3. Write down the examples of linear data structure & Non-linear data structure.
4. What are the operations can implement on stack?
5. Explain recursion using stack.
6. Explain how a string can be reversed using stack.
7. How does a stack similar to list? How it is different?
8. List the advantages and disadvantages of postfix and prefix expression over infix expression.
