

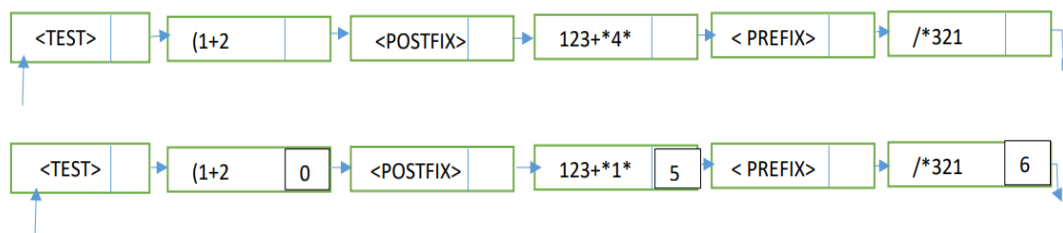
IA-4

Written Assignment

Answer all the questions in a A4 sheets, scan it and submit(Upload) in LMS on or before 30-10-2024.

Student should do the following:

1. First execute the program
 2. Write complete executable code on the A4 sheets.
 3. Put the screen shot of the sample output at the end of the code.
 4. Scan this and upload as a single pdf file
-
1. Write a menu-driven C program to perform expression processing on a set of expressions stored in a doubly linked list. The information must be stored as shown in Fig. Here, any keyword written within an angular bracket is called a 'tag'. The tag provides the meta information. There are three tags: <TEST>, <POSTFIX>, and <PREFIX>. With these details, implement the following functionalities:
 - A. Write a function Insert() to a node at the beginning of a linked list. There are 2 types of nodes: a tag node and an expression node. Each node will have a string and result field. The result field in tag node is not used, while in expression node, it is used to store the result of expression evaluation. Read the information from the user in the main() function and pass it to the Insert() function to create a list of expressions. Using a function Display(), display the list.
 - B. Write a function CheckExpression() to check whether given expression is a valid (balanced parentheses). Write another function Compute() to compute the value of the postfix/prefix expression. Using these two functions, process the linked list representing expressions and update the result field of each expression. Display the resultant list.



2. You are tasked with designing a file system hierarchy for an operating system. The file system stores two types of entries: files and directories. Directories can contain both files and other directories (nested), while files contain only data. Your system must support the following operations:

- A. Create a new file or directory in a given path.
- B. Delete a file or directory (if a directory is deleted, all its contents should be recursively deleted).
- C. List the contents of a directory (both files and subdirectories).
- D. Move a file or directory from one location to another.
- E. Find a file or directory by its name, searching recursively across all directories.
- F. Calculate the total size of a directory (sum of all file sizes inside it, including nested directories).

The file system should be represented using a tree data structure, where directories are internal nodes and files are leaf nodes.