Practical Work N°= 3 : Stack Data Structures

Objectives :

- Implement a stack using a static array.

- Implement a stack using a linked list.

- Practice the fundamental operations: push, pop, top, isEmpty, isFull.

- Use stacks to solve algorithmic problems involving expressions and number systems.

Part 1 – Static Stack (Array Implementation)

A stack is represented using: An array T of fixed size MAX and An integer top indicating the index of the last pushed element (lenght)

Tasks to do :

1. Define the structure of a static stack.

   o Write the primitive functions: init : initialize the stack, isEmpty, isFull, push, pop, top

2. Write a program that tests all the operations:

   o Push several elements, and Pop elements Checking overflow and underflow cases

Part 2 – Dynamic Stack (Linked List Implementation)

A stack node contains: An integer valueand A pointer to the next node

Tasks to do :

1. Define the structure of a stack node and a stack pointer.

2. Write the primitive functions: init : initialize the stack, isEmpty, isFull, push, pop, top

Part 3 – Applications Using Stacks :

Write the programs that :

1) Reverse an array :

Read an array of N integers **and d**isplay the reversed array (using a stack)

2) Check balanced parentheses :

Write a function: **parenthesesOK(string exp)** ; The function returns true if all parentheses are correctly balanced, otherwise false.

Practical Work N°= 3 : Stack Data Structures

Examples: (a+(b*c)) → valid   (a+b*(c-d) → invalid

3)Convert an infix expression to postfix by Using a stack to store operators and handle parentheses ; Operands are digits 0–9 and Operators: + and * only

then Evaluate the postfix expression

Examples:

- (a + b) * c → a b + c *

- a + b * c - d → a b c * + d -

Part 4 – Optional Mini-Project (Bonus)

Create a simple text editor with two stacks:

- UNDO stack

- REDO stack

Possible actions:

- Add a character

- Undo last operation (move character to REDO stack)

- Redo (move character back from REDO to text)

The program displays the current content after each action.