Term Work - 3

Problem Definition

A calculator needs to evaluate a postfix expression. Develop & execute a program in C using a suitable data structure to evaluate valid postfix expression. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands & binary arithmetic operators. The arithmetic operators are +, -, * & /.

Aim:

Aim of this TW is to learn the implementation of Stacks in solving problems.

Theory:

* Stacks : stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO or FIFO

* Mainly following basic operations are performed in Stack
  - → Push : Adds an item in stack. If stack is full then it is said to overflow
  - → Pop : Removes an item from stack. If stack is empty, then stack is said to underflow
  - → Peek : Returns top element of stack.
  - → isEmpty : Returns true if the stack is empty

Program:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct stack {
    int capacity, top;
    int *a;
};
int isEmpty(struct stack *s) {
    return s->top == s->capacity;
}
void push(struct stack *s, int op) {
    s->a[++s->top] = op;
}
int pop(struct stack *s) {
    if(!isEmpty(s)){
        return s->a[s->top--];
    }
}
int peek(struct stack *s) {
    if(!isEmpty(s){
        return s->a[s->top];
    }
}
void postFixToInfix(char * exp) {
    int i, op1=0, op2=0, result=0, x=0;
```

```
struct stack *s = (struct stack*)malloc(sizeof(struct stack));
s->top = -1;   s->capacity = strlen(exp);
s->a = (int*)malloc(sizeof(struct int) * s->capacity);
for(i=0; i<s->capacity; i++){
    if(exp[i]>='0' && exp[i]<='9'){
        x = exp[i] - '0';
        push(s, x);
    }
    else{
        op2 = pop(s);
        op1 = pop(s);
        switch(exp[i]){
            case '+': result = op1 + op2;
            break;
            case '-': result = op2 - op1;
            break;
            case '*': result = op2 * op1;
            break;
            case '/': result = op2 / op1;
            break;
        }
        push(s, result);
    }
result = pop(s)
printf("\n Answer : %d", result);
}
```

```
int main()
{
    char exp[100];

    printf("\n Enter a postfix expression: ");
    scanf("%s", exp);

    postFixToInfix(exp);
}
```

References

Books:

* Richard F Gilberg, Behrouz A Fouruzan, Data Structures: A Pseudo Code Approach with C, Cengage 2007.

* Horowitz, Sahni, Anderson-Freed, Fundamentals of Data Structures in C, Universe Press 2nd Edition.

E-Resources:

* https://geeksforgeeks.org/

Conclusion.

In this TW, I learnt about stacks, basic operations of stacks & their implementation to solve problems. We also learned basic problem solving techniques & programming paradigms.