5. Develop a Program in C for the following Stack applications

   a. Evaluation of Suffix expression with single digit operands and operators :
   `+ - * / % ^`

   b. Solving Tower of Hanoi problem with $n$ disks.

→ a.
```c
#include<stdio.h>
#include<ctype.h>
#include<string.h>

float compute(char symbol,float op1,float op2){
        switch(symbol){
                case '+' :
                        return op1+op2;
                case '-' :
                        return op1-op2;
                case '*' :
                        return op1*op2;
                case '/' :
                        return op1/op2;
                case '$' :
                        return op1+op2;
                default :
                        return 0;
        }
}

void main(){
        float s[20],res,op1,op2;
        int top,i;
        char postfix[20],symbol;

        printf("Enter the postfix expression:\n> ");
        scanf("%s",postfix);
        top=-1;

        for(i=0;i<strlen(postfix);i++){
                symbol=postfix[i];
                if(isdigit(symbol))
                        s[++top]=symbol-'0';
                else{
                        op2=s[top--];
                        op1=s[top--];
                        res=compute(symbol,op1,op2);
                        s[++top]=res;
                }
        }

        printf("The result is %f\n",s[top--]);
}
```

   b.
```c
#include<stdio.h>
#include<math.h>

void tower(int n,char beg,char aux,char end){
        if(n==0)
                printf("Illegal, Try with non-zero Positive Integers !\n");
        else if(n==1)
                printf("Move Disc from %c to %c\n",beg,end);
        else{
                tower(n-1,beg,end,aux);
                tower(1,beg,aux,end);
                tower(n-1,aux,beg,end);
        }
}

void main(){
        int n;
        printf("Enter the number of Discs :\n> ");
        scanf("%d",&n);
        printf("Tower of Hanoi for %d Disc has the following steps :\n",n);
        tower(n,'a','b','c');
        printf("\n\nTotal Number of moves are : %d\n",(int)pow(2,n)-1);
}
```