```
/*5. Design, Develop and Implement 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*/

#define SIZE 50
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include <ctype.h>

int stk[SIZE];
int top = -1;

void push(int elem)    /* Function for push operation */
{
      stk[++top] = elem;
}

int pop()                    /* Function for pop operation */
{
      return stk[top--];
}

void postfix_eval(char *postfix) /* Function for converting infix to
postfix */
{
    char sym;
      int i = 0, op1, op2;
      top = -1;
      while ((sym = postfix[i++]) != '\0')
      {
            if (isdigit(sym))            /*if symbol is operand (number)*/
                  push(sym - '0');  /* Push after converting String to
ASCII*/
            else
            {                       /*If Operator, pop two  operands */
                  op2 = pop();
                  op1 = pop();
                  switch (sym)
                  {
                  case '+':push(op1 + op2);
                        break;
                  case '-':push(op1 - op2);
                        break;
                        case '*':push(op1 * op2);
                        break;
                  case '/':push(op1 / op2);
                        break;
                  case '%':push(op1 % op2);
                        break;
                  case '^':push(pow(op1, op2));
                        break;
                  default: printf("Enter a valid operator/n");
                        exit(0);
                        break;
```

```
                }
            }
      }
      printf("Result of Postfix expression is %d", pop());
}

void tower_of_hanoi(int n, char pegA, char pegB, char pegC)    /*
Function for tower of hanoi solution */
{
      int ele;
      if (n == 1)
                                          /*for one disk*/
      {
            printf("\n Move disk 1 from peg %c to peg %c", pegA, pegB);
            /*move from pegA to pegB*/
      }
      else
      {
            tower_of_hanoi(n - 1, pegA, pegC, pegB);
            tower_of_hanoi(1, pegA, pegB, pegC);
            tower_of_hanoi(n - 1, pegC, pegB, pegA);
      }

}

int main()
{
      char postfix[30];
      int ch, n, i;
      while (1)
      {
            printf("\n\n*********Stack Applications Menu*********\n\n");
            printf("1. Evaluation of Suffix Expression\n");
            printf("2. Solve Tower of hanoi\n");
            printf("3. Exit\n");
            printf("Enter your choice:\n");
            scanf("%d", &ch);
            switch (ch)
            {
            case 1: printf("Enter a valid Postfix expression with single
digit operands\n");
                  scanf("%s", postfix);
                  postfix_eval(postfix);
                  break;
            case 2: printf("Enter number disks\n");
                  scanf("%d", &n);
                  printf("The sequence of moves involved in the Tower of
Hanoi are :\n");
                  tower_of_hanoi(n, 'A', 'B', 'C');
                  break;
            case 3: exit(0);
            default: printf("Enter the valid choice\n\n");
                  break;
            }
      }
      return 0;
}
```