LAB PROGRAMS
Data Structure and Applications

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

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
void evaluate(char []);
void tower(int,char S,char D,char A);
int stack[20];
int top=-1;
void push(int x)
{
stack[++top]=x;
}
int pop()
{
return stack[top--];

}
int main()
{
int ch,n;
char exp[100],S,D,A;
while(1)
{
printf(" Application of stack\n");
printf("\n 1: Evaluation of postfix expression\n");
printf("\n 2. Tower of hanoi\n");
printf(" 3.exit\n");
printf(" enter ur choice\n");
```

```
scanf("%d", &ch);
switch(ch)
{
 case 1: printf(" Enter a valid postfix Expression\n");
        scanf("%s", exp);
        evaluate(exp);
        break;
 case 2:printf(" Enter number of disks\n");
      scanf("%d", &n);
      tower(n,'S','D','A');
     printf("Number of moves is %d",(int)pow(2,n)-1);
      break;
 case 3: exit(0);
 default: printf(" invalid choice");
}
}
}
void evaluate(char exp[100])
{

char symb;
int n1,n2,n3,num,i;
for(i=0;exp[i]!='\0';i++)
{
symb=exp[i];
if(isdigit(symb))
{
num=symb-48;
push(num);
}
else
{
n1=pop();
n2=pop();
switch(symb)
{
case '+':n3=n1+n2;
      break;
case '-':n3=n2-n1;
      break;
case '*':n3=n1*n2;
      break;
case '/':n3=n2/n1;
      break;
case '%':n3=n2%n1;
      break;
}
push(n3);
}
}
printf("The result of  postfix expression %s is =%d\n", exp,pop());

}
void tower(int n, char S, char D, char A)
{
   if(n==0)
   {
      return;
   }
  if(n==1)
  {
     printf("Move disk 1 from %c to %c\n",S,D);
     return;
  }
  else
  {

     tower(n-1,S,A,D);
     printf("Move disk %d from %c to %c\n", n,S,D);
     tower(n-1,A,D,S);
```

```
    }
}
```