**DATA STRUCTURES LABORATORY**

**PROGRAM 5**

1. **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**

Program:

**A.**

#include<stdio.h>

#include<math.h>

#include<string.h>

#include<ctype.h>

double compute(char symbol,double op1,double op2);

void main()

{

double s[20],res,op1,op2;

int top=-1,i;

char postfix[20],symbol;

printf("Enter the Postfix Expression\n");

scanf("%s",postfix);

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;

}

}

res=s[top--];

printf("The result %f \n",res);

}

double compute(char symbol,double op1,double op2)

{

switch(symbol)

{

case '+':return op1+op2;

case '-':return op1-op2;

case '/':return op1/op2;

case '\*':return op1\*op2;

/\*case '^':

case '$':return (pow(op1,op2)) ;\*/

}

}

**Output:**

Enter the Postfix Expression

23+4\*

The result 20.000000

Program:

**B.**

#include<stdio.h>

#include<stdlib.h>

void TOH( int n, char A, char B, char C);

void main()

{

int n;

printf("Enter number of rings:\n");

scanf("%d", &n);

TOH(n, 'A', 'C', 'B');

}

void TOH( int n, char A, char B, char C)

{

if(n==1)

printf("Move from %c to %c\n",A, B);

else

{

TOH(n-1, A, C, B);

TOH(1, A, B, C);

TOH(n-1, C, B, A);

}

}

**Output:**

Enter number of rings:

3

Move from A to C

Move from A to B

Move from C to B

Move from A to C

Move from B to A

Move from B to C

Move from A to C