**Design, Develop and Implement a Program 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.

**Student Name:- Neha Sharma** **UID: 20BCS4576**
**Branch: CSE-IOT** **Section/Group: A**
**Semester: 3rd** **Date of Performance:27-09-2021**
**Subject Name: DATA STRUCTURES LAB**
**Subject Code: 21O-20CSP-236_20BIT-1_A**

## 1. Aim/Overview of the practical:

Design, Develop and Implement a Program 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.

## 2. Task to be done:

a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^
b. Solving Tower of Hanoi problem with n disks.

## 3. Algorithm/Flowchart:

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

**Step 1 While reading the expression from left to right.**
**step 2 push the element in the stack if it is an operand.**
**Step 3 Pop the two operands from the stack .**
**step 4 if the element is an operator and then evaluate it.**
**Step 5 Push back the result of the evaluation.**
**Step 6 Repeat it till the end of the expression .**

**Solving Tower of Hanoi problem with n disks.**

**1.Start**
**2.Define function**
**3.Declare variable n**
**4.Enter the number of disc**
**5.Call the function**
**6.Stop**

**4. Steps for experiment/practical:**

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

```
#include<stdio.h>

#include<math.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 '$':
 case '^': return pow(op1,op2);
 default : return 0;
 }
}
void main()
{
float s[20], res, op1, op2;
int top, i;
char postfix[20], symbol;
printf("\nEnter 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;
 }
}
res = s[top--];
printf("\nThe result is : %f\n", res);
}
```

## Solving Tower of Hanoi problem with n disks.

```
#include<stdio.h>
#include<conio.h>
void towerOfHanoi(int n, char from_rod, char to_rod, char aux_rod)
{
```

```
 if (n == 1)
 {
   printf("\n Move disk 1 from rod %c to rod %c", from_rod, to_rod);
   return;
 }
 towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
 printf("\n Move disk %d from rod %c to rod %c", n, from_rod, to_rod);
 towerOfHanoi(n-1, aux_rod, to_rod, from_rod);
}
 int main()
{
 int n = 4; // Number of disks
 towerOfHanoi(n, 'A', 'C', 'B'); // A, B and C are names of rods
 return 0;
}
```

**5. Output: Image of sample output to be attached here**

1.

```
Enter the postfix expression:
234*+

The result is : 14.000000


...Program finished with exit code 0
Press ENTER to exit console.
```

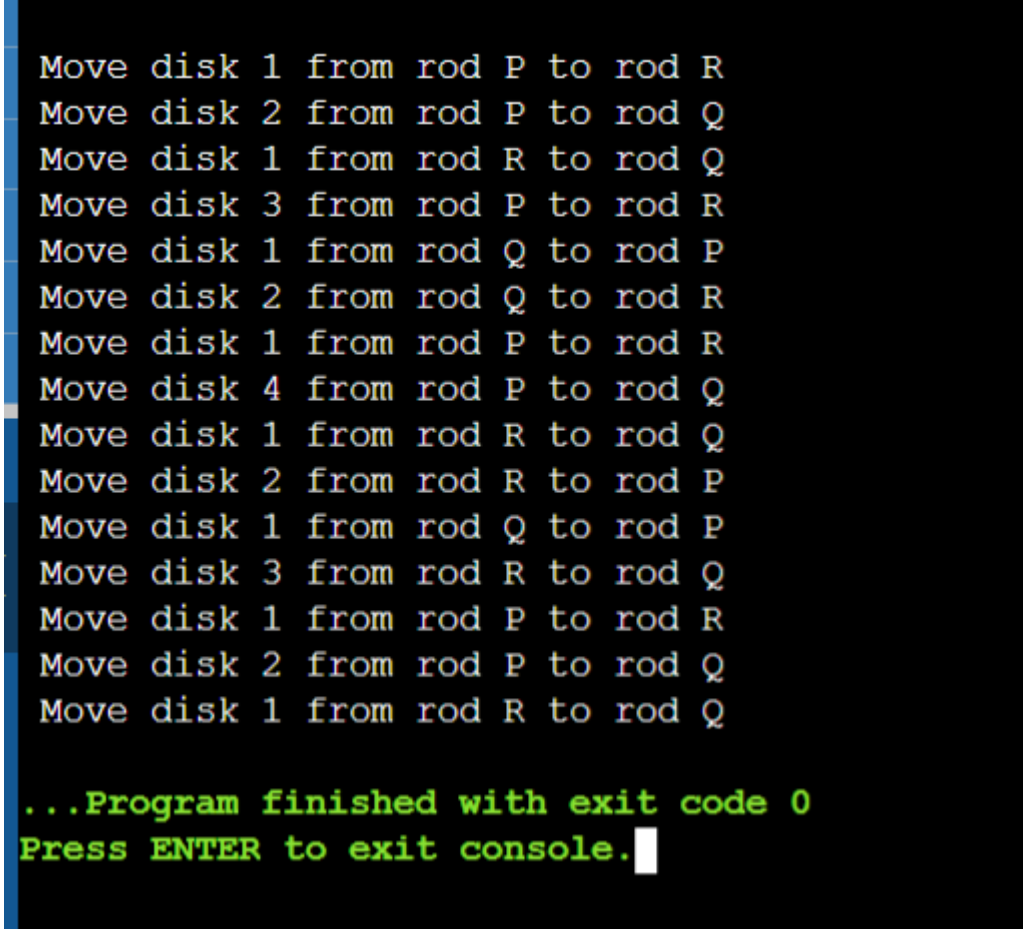
1.

```
Enter the postfix expression:
234*+*+

The result is : 56.000000


...Program finished with exit code 0
Press ENTER to exit console.
```

**2.**


Move disk 1 from rod P to rod R
Move disk 2 from rod P to rod Q
Move disk 1 from rod R to rod Q
Move disk 3 from rod P to rod R
Move disk 1 from rod Q to rod P
Move disk 2 from rod Q to rod R
Move disk 1 from rod P to rod R
Move disk 4 from rod P to rod Q
Move disk 1 from rod R to rod Q
Move disk 2 from rod R to rod P
Move disk 1 from rod Q to rod P
Move disk 3 from rod R to rod Q
Move disk 1 from rod P to rod R
Move disk 2 from rod P to rod Q
Move disk 1 from rod R to rod Q
...Program finished with exit code 0
Press ENTER to exit console.

## Learning outcomes (What I have learnt):

1. **Design, Develop and Implement a Program for the Stack Applications.**
2. **Solving Tower of Hanoi problem with n disks.**
3. **Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^**

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
|---|---|---|---|
| 1. | | | |
| 2. | | | |

| | | | |
|---|---|---|---|
| 3. | | | |
| | | | |