

```
/* DS Lab 5
```

```
Develop a Program in C for the following Stack Applications
```

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

```
A
```

```
b. Solving Tower of Hanoi problem with n disks
*/
```

```
#include<stdio.h>
```

```
#include<math.h>
```

```
double compute (double operand1, char operator, double operand2)
```

```
{
```

```
    switch(operator)
```

```
    {
```

```
        case '+' : return operand1 + operand2;
```

```
        case '-' : return operand1 - operand2;
```

```
        case '*' : return operand1 * operand2;
```

```
        case '/' : return operand1 / operand2;
```

```
        case '^' :
```

```
        case '$' : return pow(operand1, operand2);
```

```
    }
```

```
}
```

```
double evaluate (char postfix[])
```

```
{
```

```
    int i, top = -1;
```

```
    double stack[20], operand1, operand2;
```

```
    for(i = 0; postfix[i]!='\0'; i++)
```

```
    {
```

```
        if(postfix[i] >= '0' && postfix[i] <= '9')
```

```
            stack[++top] = postfix[i] - '0';
```

```
        else
```

```
        {
```

```
            operand2 = stack[top--];
```

```
        operand1 = stack[top--];
        stack[++top] = compute(operand1, postfix
                               [i], operand2);
    }

    return stack[top--];
}

void main()
{
    char postfix[20];
    double result;

    printf("Enter Postfix Expression : ");
    scanf("%s", postfix);

    result = evaluate(postfix);
    printf("Evaluation of Postfix Expression =
    %lf\n", result);
}
```

/\*

Output:

Enter Postfix Expression : 632-5\*+2^3+

Evaluation of Postfix Expression = 124.000000

Enter Postfix Expression : 12+3-21+3\$-

Evaluation of Postfix Expression = -27.000000

...Program finished with exit code 0

Press ENTER to exit console.

\*/

## //b. Solving Tower of Hanoi problem with n disks

```
#include <stdio.h>
```

```
// Recursive function to solve Tower of Hanoi  
problem
```

```
void towerOfHanoi(int n, char source, char temp,  
char destination)  
{  
    if (n == 1)  
    {  
        printf("Move Disk 1 from %c to %c\n", source  
            , destination);  
        return;  
    }  
  
    // Move n-1 disks from from_rod to aux_rod  
    using to_rod  
    towerOfHanoi(n - 1, source, destination, temp);  
    printf("Move Disk %d from %c to %c\n", n, source  
        , destination);  
  
    // Move n-1 disks from aux_rod to to_rod using  
    from_rod  
    towerOfHanoi(n - 1, temp, source, destination);  
}
```

```
void main()  
{  
    int n;  
  
    printf("Enter the number of Disks: ");  
    scanf("%d", &n);  
  
    printf("Solution for Tower of Hanoi with %d  
    disks:\n", n);  
    towerOfHanoi(n, 'A', 'C', 'B'); // A, B, C are  
    names of rods
```

```
}
```

```
/*
```

```
Output :
```

```
Enter the number of Disks: 1
```

```
Solution for Tower of Hanoi with 1 disks:
```

```
Move Disk 1 from A to B
```

```
Enter the number of Disks: 2
```

```
Solution for Tower of Hanoi with 2 disks:
```

```
Move Disk 1 from A to C
```

```
Move Disk 2 from A to B
```

```
Move Disk 1 from C to B
```

```
Enter the number of Disks: 3
```

```
Solution for Tower of Hanoi with 3 disks:
```

```
Move Disk 1 from A to B
```

```
Move Disk 2 from A to C
```

```
Move Disk 1 from B to C
```

```
Move Disk 3 from A to B
```

```
Move Disk 1 from C to A
```

```
Move Disk 2 from C to B
```

```
Move Disk 1 from A to B
```

```
*/
```