Experiment 2

Student Name: Jitesh Kumar **UID: 20BCS2334**
Branch: CSE **Section/Group: 20BCS-WM-903/A**
Semester: 5th
Subject Name: DAA Lab **Subject Code:** 21-CSP-312

1. Aim/Overview of the practical:

Code implement power function in O(logn) time complexity.

2. Task to be done/ Which logistics used:

To find Power of a number.

3. Algorithm/Flowchart (For programming based labs):

Step1: Take x and n input.
Step2: Calculate pow(x, n) method check base condition if n==0 return 1 check base condition if n==1 return x recursively callpow(x,n-1) and go to step 2;
Step 3: Print result.

4. Steps for experiment/practical/Code:

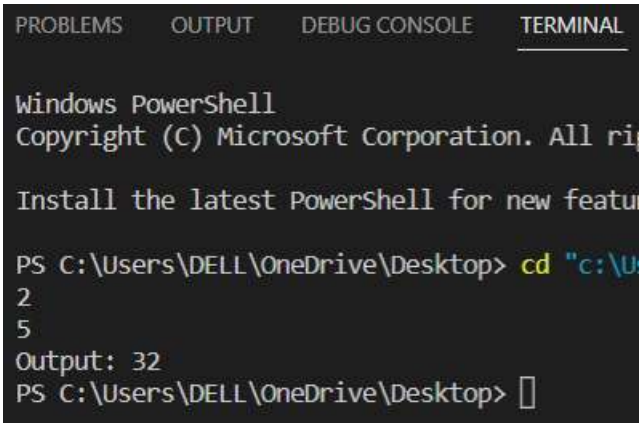
```
#include<bits/stdc++.h>
using namespace std;
double power(double n,int x)
{
if(x==0)
   return 1;

double temp=power(n,x/2);
if(x%2==0)
{
   return temp*temp;
}
else if(x>0)
{
   return n*temp*temp;
}
else
   return (temp*temp)/n;
}
int main()
```

```
{
   double n;
   int x;
   cin>>n>>x;
   double ans=power(n,x);
   cout<<"Output: "<<ans<<endl;
}
```

5. Observations/Discussions/ Complexity Analysis:

Time complexity of finding power of a number using recursion is O(log n).

6. Result/Output/Writing Summary:



Learning outcomes (What I have learnt):

1. To know to calculate power of a function.
2. To learn how to use recursion for solving problems.