**Course Name: DAA Lab** **Course Code: 21ITH-311/21CSH-311**

# Experiment: 1.2

**Aim:** Develop a program for implementation of power function and determine that complexity should be O(log n)

**Objectives**: To understand power function

**Input/Apparatus Used:** STL commands are used using C++ language

**Procedure/Algorithm:**

Step1: Start

Step2: Declare the variables int num and int power

Step3: Create the function and pass the parameters int num and int power

Step4: In function check the power==0 return and if power==1 return num

Step5: Call the recursive function func(num,power-1) and check for total number of power

Step6: Return the recursive function with the multiplication of num

Step7: End

## Sample Code:

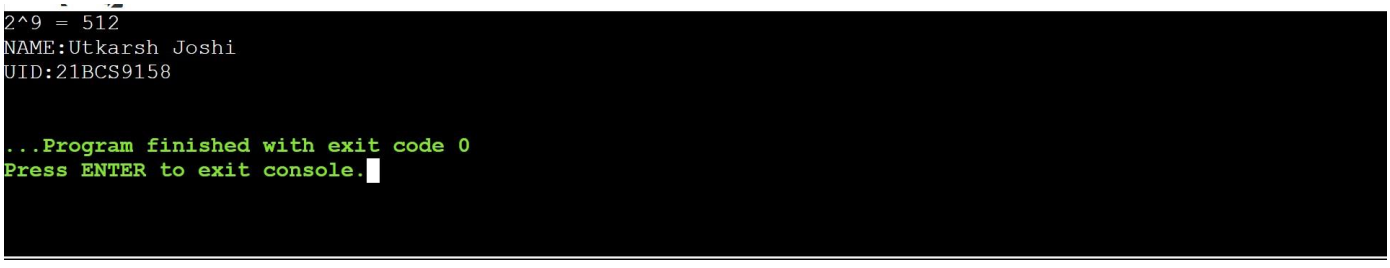
```
#include <iostream>

int power(int x, int n)

{     if (n
== 0) {
return 1;
}     if (n
== 1) {
return x;
    }     if (n % 2 == 0) {
return power(x * x, n / 2);
    } else {
        return x * power(x, n - 1);
    }

} int main() {     int x =
2;     int n = 9;     int
result = power(x, n);
    std::cout << x << "^" << n << " = " << result << std::endl;
std::cout<<"NAME:Utkarsh Joshi "<<std::endl;
    std::cout<<"UID:21BCS9158" <<std::endl;
    return 0;
```

## Observations/Outcome :




**Time Complexity:** O(logn)