

Course Name: DAA Lab

Course Code: 21ITH-311/21CSH-311

**Student Name:** Amit Kumar  
**Branch:**BE-CSE  
**Semester:**5  
**Subject Name:**DAA

**UID:** 21BCS9056  
**Section/Group:** 21BCS\_IOT\_644/B  
**Date of Performance:**22-08-2023  
**Subject Code:** 21CSH-311

## Experiment 1.2

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

### Procedure/Algorithm:

- Step 1: Start  
Step 2: Define the power function with parameters base and exponent.  
Step 3: Base case – Exponent is 0
- If true return 1
  - If false, continue to next step
  - Step 4: If exponent is even
  - If true, calculate ans as the result of calling the power function with parameters base and exponent / 2.
  - Return ans\* ans as the result.
- Step 5: If exponent is odd
- If the exponent is odd, calculate ans as the result of calling the power function with parameters base and (exponent - 1) / 2.
  - Return base \* ans\* ans as the result.
- Step 6: Store the output return by function in result. Display result. Step 7: End

### Sample Code:

```
#include<iostream> using namespace  
std; int powerCalc(int base,int  
exponent){ if(exponent==0){  
return 1;  
}  
if(exponent%2==0){  
int ans=powerCalc(base,exponent/2);  
return ans*ans;
```



Course Name: DAA Lab

Course Code: 21ITH-311/21CSH-311

```
else{
    int ans=powerCalc(base,exponent/2);
    return base*ans*ans;
}

int main(){    int
base,exponent;
cout<<"Enter the base: ";
cin>>base;
    cout<<"\nEnter the exponent: ";
cin>>exponent;

    int result= powerCalc(base,exponent);

    cout<<"\n"<<base<<" to the power "<<exponent<<" is: "<<result;
}
```

**Observations/Outcome:**

---

The screenshot displays an online C++ compiler interface. The top section shows the source code for a program named `main.cpp`. The code implements a recursive function `powerCalc` to calculate the power of a base raised to an exponent. The function uses a base case where the exponent is 0, returning 1, and a recursive case where the exponent is greater than 0, dividing the exponent by 2 and multiplying the base by the result of the recursive call. The `main` function prompts the user for a base and an exponent, calls `powerCalc`, and prints the result.

```
1 #include<iostream>
2 using namespace std;
3 int powerCalc(int base,int exponent){
4     if(exponent==0){
5         return 1;
6     }
7     if(exponent%2==0){
8         int ans=powerCalc(base,exponent/2);
9         return ans*ans;
10    }
11    }
12    else{
13        int ans=powerCalc(base,exponent/2);
14        return base*ans*ans;
15    }
16 }
17 int main(){
18     int base,exponent;
19     cout<<"Enter the base: ";
20     cin>>base;
21     cout<<"\nEnter the exponent: ";
22     cin>>exponent;
23     int result= powerCalc(base,exponent);
24     cout<<"\n"<<base<<" to the power "<<exponent<<" is: "<<result;
25 }
```

The bottom section shows the program's execution output. It displays the prompts and user input: "Enter the base: 2" and "Enter the exponent: 5". The output then shows the result: "2 to the power 5 is: 32". The program concludes with the message "...Program finished with exit code 0" and a prompt to "Press ENTER to exit console."

## Time Complexity:

Time complexity is  $O(\log n)$

Space complexity is  $O(1)$

Name: Amit Kumar

UID: 21BCS9056