Student Name: Amit Kumar **UID:** 21BCS9056
Branch:BE-CSE **Section/Group:** 21BCS_IOT_644/B
Semester:5 **Date of Performance:**22-08-2023
Subject Name:DAA **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;
```

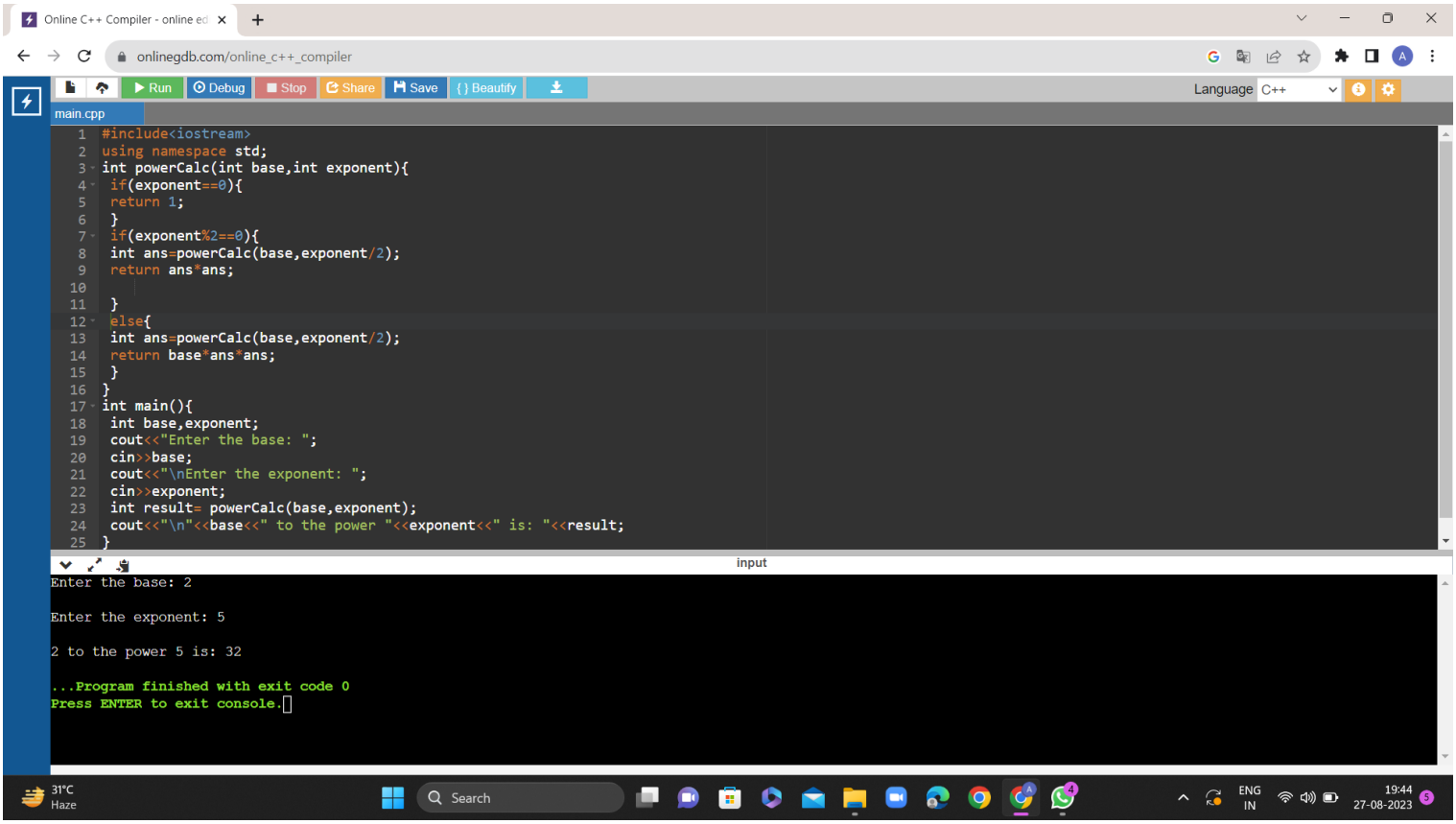
```
    else{
            int ans=powerCalc(base,exponent/2);
            return base*ans*ans;
        }
    }

    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:



Time Complexity:

Time complexity is O(log n)
Space complexity is O(1)

Name: Amit Kumar **UID: 21BCS9056**