Experiment Title 2

Student Name: Pranjal Kumar **UID:** 20BCS3504
Branch: CSE **Section/Group:** 607 B
Semester: 5th **Date of Performance:** 25/08/22
Subject Name: DAA **Subject Code:** 20CSP-312

1. Aim/Overview of the practical:

Code to implement power function in O(log n) time complexity.

2. Task to be done/ Which logistics used:

To find the power of the function with the given time complexity.

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

METHOD 1: Time complexity==O(n):

Step 1: create a function power() which takes two arguments data and the power value;

Step 2: if power value is equal to zero then return 1 (i.e data^0==1)

Step 3: if power value is equal to even recursively call the power() function until the power value doesn't become zero .

That is power(data,value/2)*power(data,value/2)

Step 4: if power value is odd then call the power() function recursively until power value becomes zero

That is : data*power(data,value/2)*power(data,value/2);

METHOD 2: Time Complexity==O(log(n))

In this method you will be calling the power function only once and store the value of power function into temporary variable and return the value of temp hence the number of call will be reduced half time for every call;

Step 1: create a function power() which takes two arguments data and the power value;

Step 2: if power value is equal to zero then return 1 (i.e data^0==1)

Step 3: create temporary variable store the value returned by power() function ;

That is temp=power(data, value/2);

Step 4: if power value is even return temp*temp;

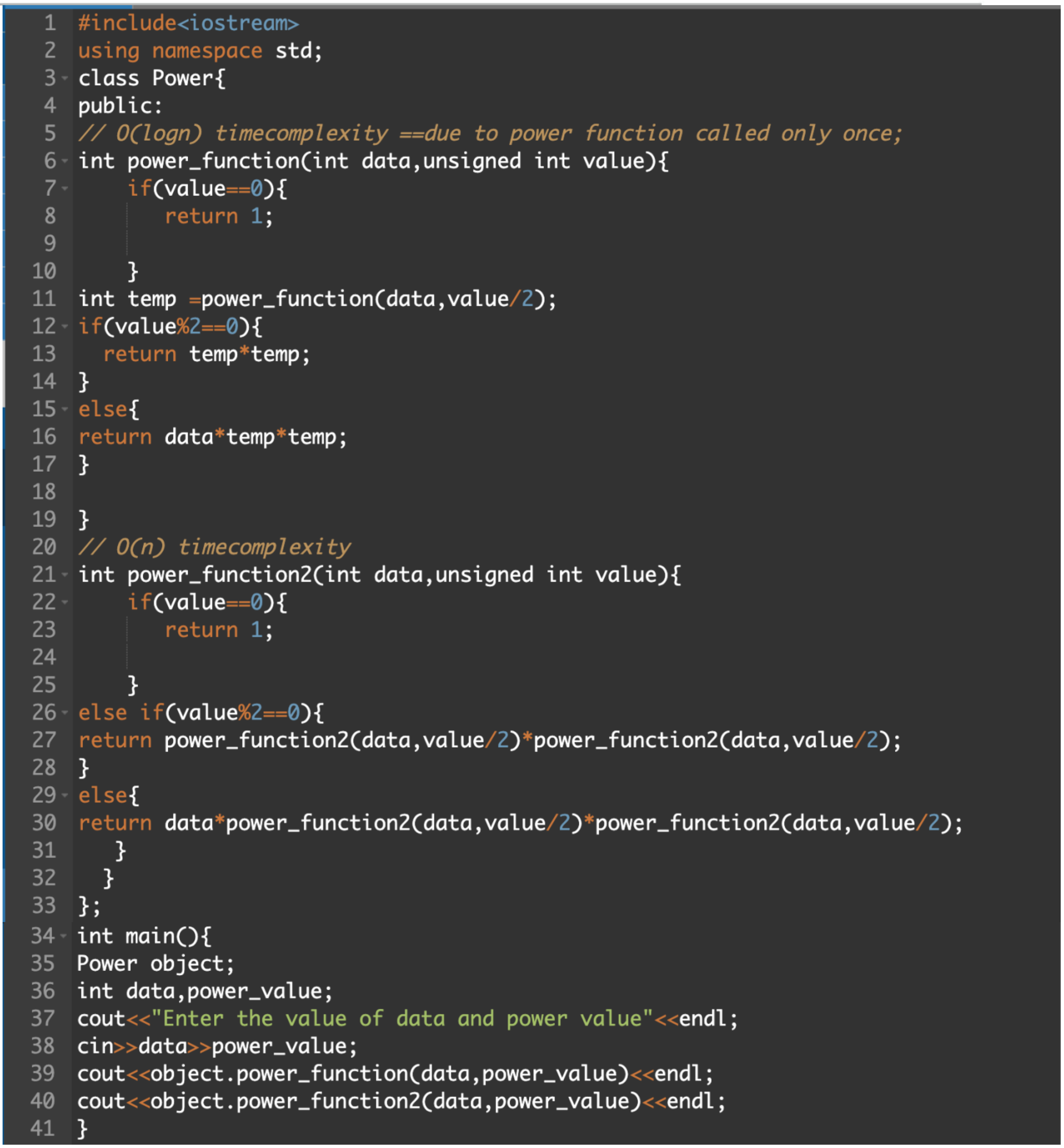
That is return temp*temp;

Step 5: if power value is odd return data*temp*temp;

That is => return (data*temp*temp);

4. Steps for experiment/practical/Code:

```
 #include<iostream>

using namespace std;

class Power{

public:

// O(logn) timecomplexity ==due to power function called only once;

int power_function(int data,unsigned int value){

    if(value==0){

       return 1;


    }

int temp =power_function(data,value/2);

if(value%2==0){

  return temp*temp;

}

else{

return data*temp*temp;
}

}

// O(n) timecomplexity

int power_function2(int data,unsigned int value){

    if(value==0){

        return 1;

    }

else if(value%2==0){

return power_function2(data,value/2)*power_function2(data,value/2);

}

else{

return data*power_function2(data,value/2)*power_function2(data,value/2);

   }

  }

};

int main(){

Power object;

int data,power_value;

cout<<"Enter the value of data and power value"<<endl;

cin>>data>>power_value;

cout<<object.power_function(data,power_value)<<endl;

cout<<object.power_function2(data,power_value)<<endl;

}
```



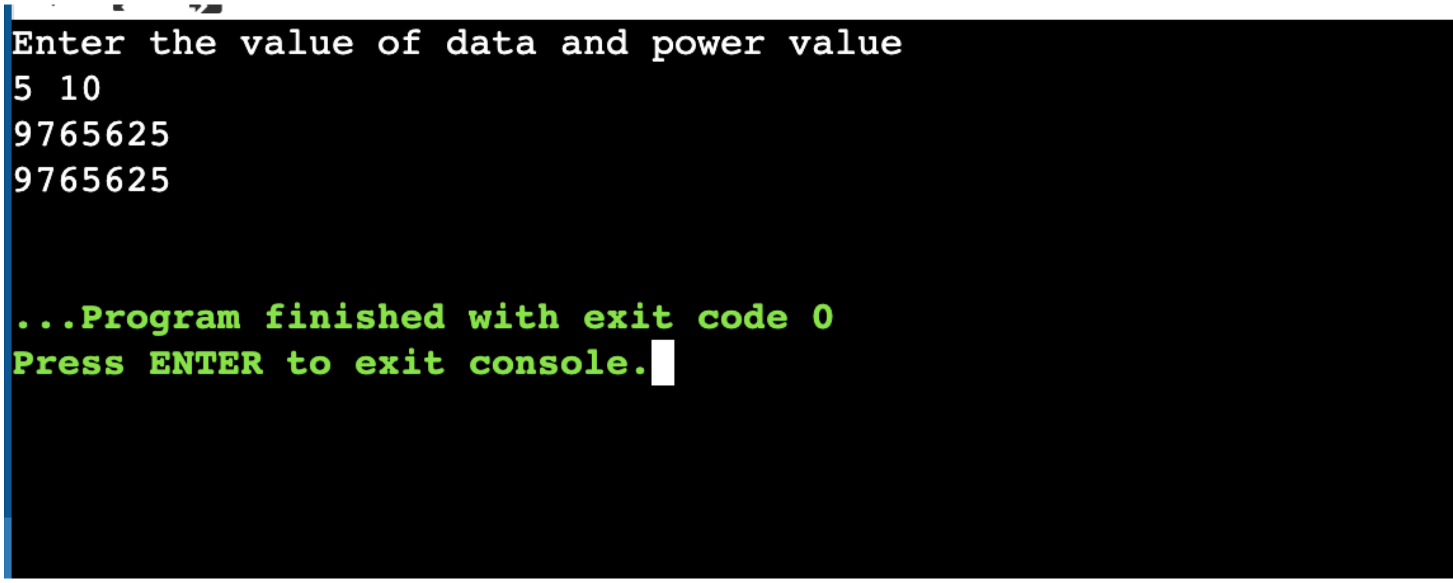
```cpp
#include<iostream>
using namespace std;
class Power{
public:
// O(logn) timecomplexity ==due to power function called only once;
int power_function(int data,unsigned int value){
    if(value==0){
       return 1;

    }
int temp =power_function(data,value/2);
if(value%2==0){
  return temp*temp;
}
else{
return data*temp*temp;
}

}
// O(n) timecomplexity
int power_function2(int data,unsigned int value){
    if(value==0){
       return 1;

    }
else if(value%2==0){
return power_function2(data,value/2)*power_function2(data,value/2);
}
else{
return data*power_function2(data,value/2)*power_function2(data,value/2);
   }
  }
};
int main(){
Power object;
int data,power_value;
cout<<"Enter the value of data and power value"<<endl;
cin>>data>>power_value;
cout<<object.power_function(data,power_value)<<endl;
cout<<object.power_function2(data,power_value)<<endl;
}
```

5. Observations/Discussions/ Complexity Analysis:

1) In method 1 with time complexity ==O(n) (linear complexity) we were calling the power function two times and also the statements inside the function was executing n times hence to reduce the number of execution of codes we created the another function which will call the power function only once and the number of statements will be reduced to half hence method 2 is efficient for large number of power value so method 2 is always preferred with time complexity O(log(N)).

2)Space complexity of both the algorithm is constant

6. Result/Output/Writing Summary:



```
Enter the value of data and power value
5 10
9765625
9765625


...Program finished with exit code 0
Press ENTER to exit console.
```

7)Learning outcomes (What I have learnt):

1) Learned about power function and how to write the code efficiently

2) Learned about Time complexity and Space complexity of power function

3) Learned how to write the code efficiently

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			