

“Experiment 1.2”

Student Name: **SUMIT KUMAR**

Branch: **CSE**

Semester: **5**

Subject Name: **Design and Analysis of Algorithms Lab**

UID: **20BCS8226**

Section/Group: **808-A**

Date of Performance: **11-08-22**

Subject Code: **20CSP-312**

1. Aim/Overview of the practical:

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

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

START

Step 1: Let x , be the number and n be the power

Step 2: $\text{temp} \rightarrow x * x$, $n = n/2$;

Step 3: Repeat step 2 while n is greater than 0.

Step 4: $\text{temp} = x^n$;

END

3. Steps for experiment/practical/Code:

```
// SUMIT KUMAR
```

```
// 20BCS8226
```

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int power(int x, int n)
```

```
{
```

```
    if(n==0)
```

```
        return 1;
```

```
int temp = power(x, n/2) * power(x, n/2);
```

```
    if(n%2==0) {
```

```
        return temp;
```

```
    }
```

```
    else return x*temp;
```

```
}
```

```
int main()
```

```
{
```

```
    int x,n;
```

```
    cin>>x>>n;
```

```
    cout<<x<<" to the power "<<n<<" is : "<<power(x,n);
```

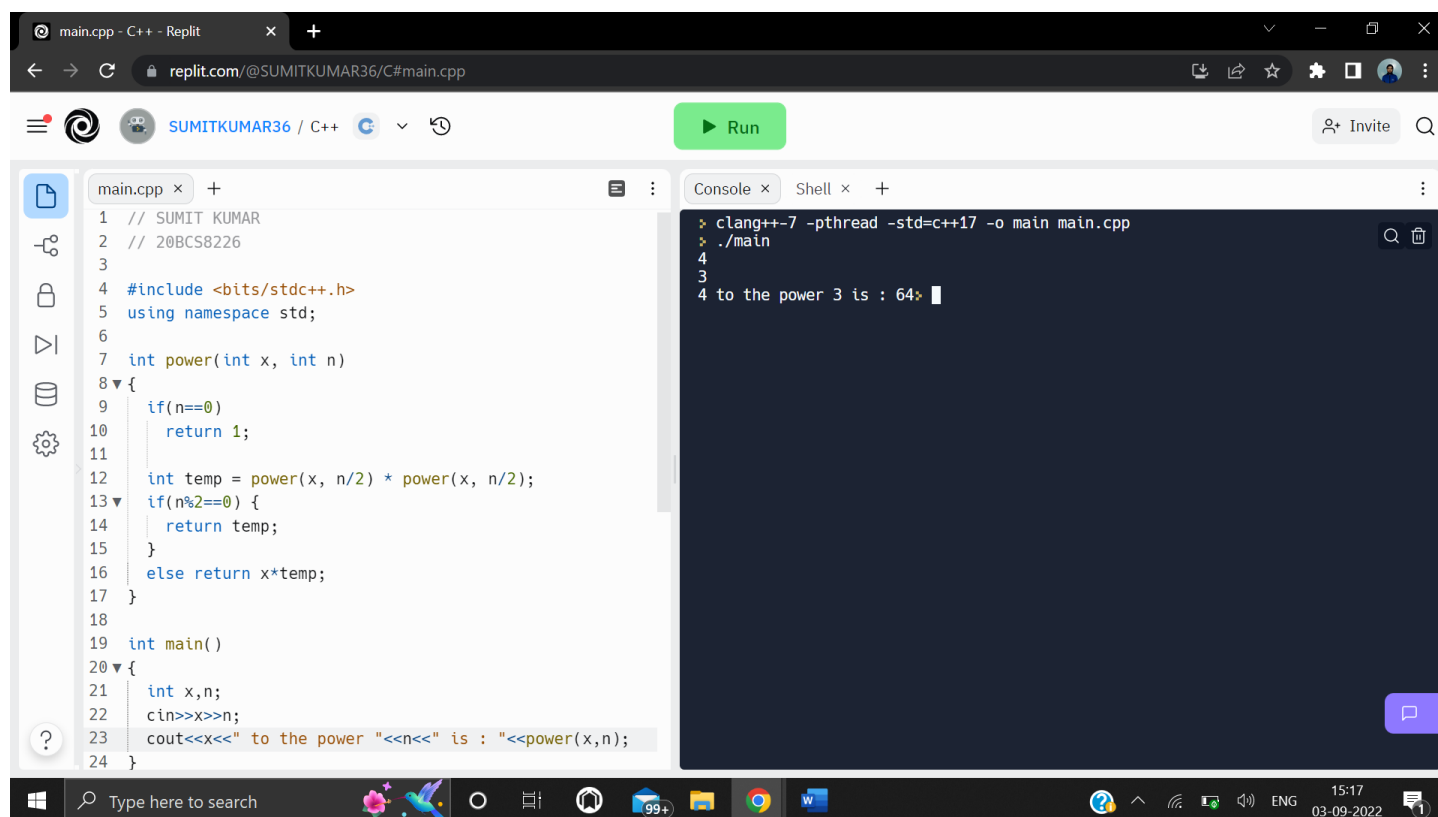
```
}
```

4. Observations/Discussions/ Complexity Analysis:

Instead of multiplying a number by itself 'n' times in a loop, we can use a recursive function, where the product is saved at each step, and we can reuse it in later calculations by divide and conquer approach.

The time complexity of this algorithm is $O(\log(n))$.

5. Result/Output/Writing Summary:



The screenshot shows a C++ program on the Replit platform. The code defines a recursive function `power` to calculate the power of a number `x` raised to `n`. The function uses a divide-and-conquer approach, handling even and odd exponents. The `main` function takes input for `x` and `n`, and prints the result. The console output shows the program running successfully, calculating $4^3 = 64$.

```
1 // SUMIT KUMAR
2 // 20BCS8226
3
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 int power(int x, int n)
8 {
9     if(n==0)
10         return 1;
11
12     int temp = power(x, n/2) * power(x, n/2);
13     if(n%2==0) {
14         return temp;
15     }
16     else return x*temp;
17 }
18
19 int main()
20 {
21     int x,n;
22     cin>>x>>n;
23     cout<<x<<" to the power "<<n<<" is : "<<power(x,n);
24 }
```

Console Output:

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
4
3
4 to the power 3 is : 64
```

6. Learning outcomes (What I have learnt):

1. Learnt faster algorithm to calculate power of a number using divide and conquer approach by recursion.
2. Learnt how to code the above algorithm in C++.
3. Learnt how to specify different cases during the flow of the program, for the recursion to work without errors, such as when the power is odd and even.

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.			