

## **WORKSHEET – 2**

**Name:** Ruchika Raj

**Section/Group:** 615 / B

**UID:** 20BCS9285

**Subject:** Design and Analysis of Algorithm Lab

**Date of Submission:** 02.09.2022

**Branch:** BE CSE

### **AIM :**

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

### **Task to be done/ Which logistics used:**

**Task:** To calculate the power of a number with time complexity  $O(\log n)$

**Logic Used:** We used the ‘Divide and Conquer’ approach to solve this question. Given two integers  $x$  and  $n$ , we have to compute  $x^n$ . We may assume that  $x$  and  $n$  are small and overflow doesn’t happen. We keep on dividing  $n$  by 2 recursively until we reach the base case i.e.  $n==1$ .

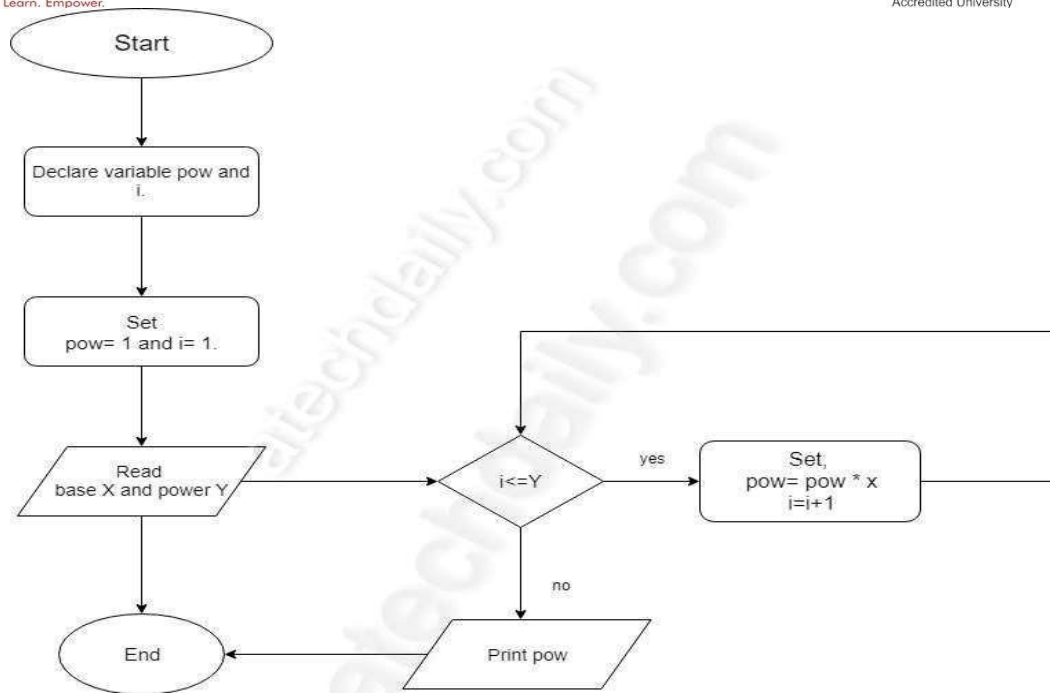
**Platform Used :** Online Compiler

### **Algorithm/Flowchart :**

1. Start the program.
2. Take two variables i.e.  $a$  and  $b$ .

3. Take the input values of  $A$  and  $B$ .
4. To retain the values of  $A$  and  $B$  let us store them in another variables named  $X$  and  $n$ . And take a variable  $result$  which is initialized to 1.
5. Now, iterate a loop till  $n$  value reaches to 0(zero).
6. For each iteration if  $n$  value is Even then we multiply  $X$  with itself then we reduce the value of  $n$  to its half.
7. And if when  $n$  value is odd then we multiply  $result$  with  $X$  value and then we reduce the value of  $n$  by 1.
8. Finally, if  $n$  value reaches to zero then we return the  $result$ .
9. End the program.

### **Flowchart :**



Protected with free version of Watermarkly. Full version doesn't put this mark

### **Pseudo Code Of The Algorithm:**

1. Let x, y be the two numbers.
2. Enter the value of x.
3. Enter the value of y.
4. Use the power function and calculate the value of function.
5. Finish.

### **CODE:**

main.cpp

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int power(int x, unsigned int y)
4  {
5      int temp;
6      if( y == 0) {
7          return 1;
8      }
9      temp = power(x, y / 2);
10     if (y % 2 == 0){
11         return temp * temp;
12     }
13     else {
14         return x * temp * temp;
15     }
16 }
17 Int main() {
18     int base;
19     int exponent;
20     cout<<"Enter the value of base: ";
21     cin>>base;
22     cout<<"Enter the value of exponent: ";
23     cin>>exponent;
24     cout<< base << " ^ " << exponent << " ="<<power(base,exponent);
25     return 0;
26 }
27

```



input

```

#include <bits/stdc++.h>
using namespace std;
int power(int x, unsigned int y)
{
    int temp;
    if( y == 0) {
        return 1;
    }
    temp = power(x, y / 2);
    if (y % 2 == 0){

```

```
        return temp * temp;
    }
    else {
        return x * temp * temp;
    }
}

Int main() {
    int base;
    int exponent;
    cout<<"Enter the value of base: ";
    cin>>base;
    cout<<"Enter the value of exponent: ";
    cin>>exponent;
    cout<< base << " ^ " << exponent << " ="<<power(base,exponent);
    return 0;
}
```

### **Observations/Discussions/ Complexity Analysis:**

A simple algorithm for this problem is to multiply 'x' by 'n' times. The time complexity of this algorithm would be  $O(n)$ . We can use the divide and conquer approach to solve this problem more efficiently.

In the dividing step, we keep dividing  $n$  by 2 recursively until we reach the base case i.e.  $n == 1$

In the combining step, we get the result, 'r', of the sub-problem and compute the result of the current problem using the two rules below:

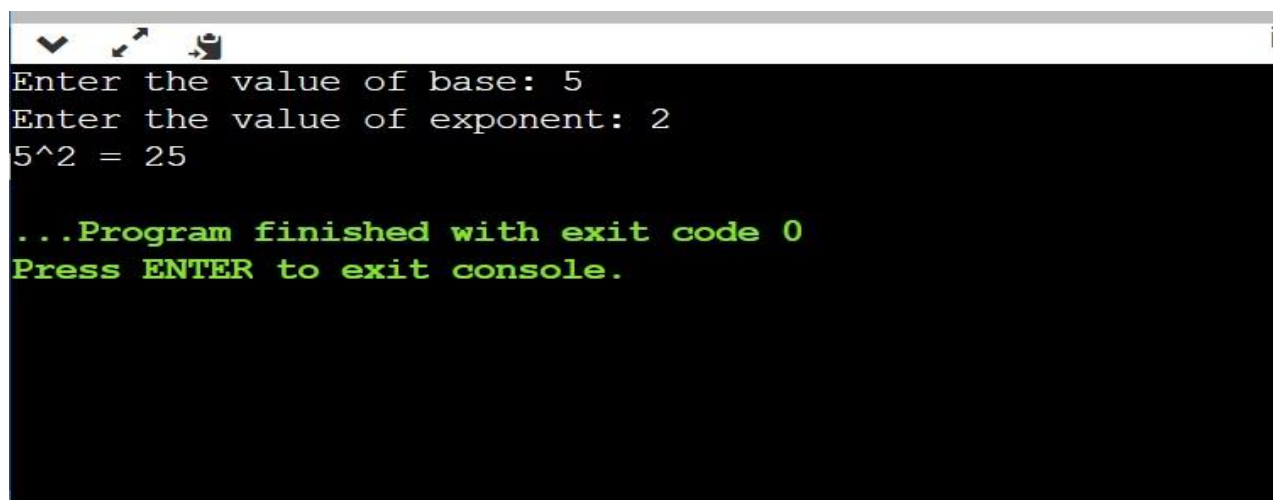
if  $n$  is even, the result is  $r * r$  (where  $r$  is the result of sub-problem)

if  $n$  is odd, the result is  $x * r * r$  (where  $r$  is the result of sub-problem)

**Time Complexity:**  $O(\log(n))$  . Here, we are reducing the value of  $n$  to its half. So, we are just iterating the loop by  $n$  value. So, time Complexity is reduced to  $O(\log(n))$  from  $O(n)$ .

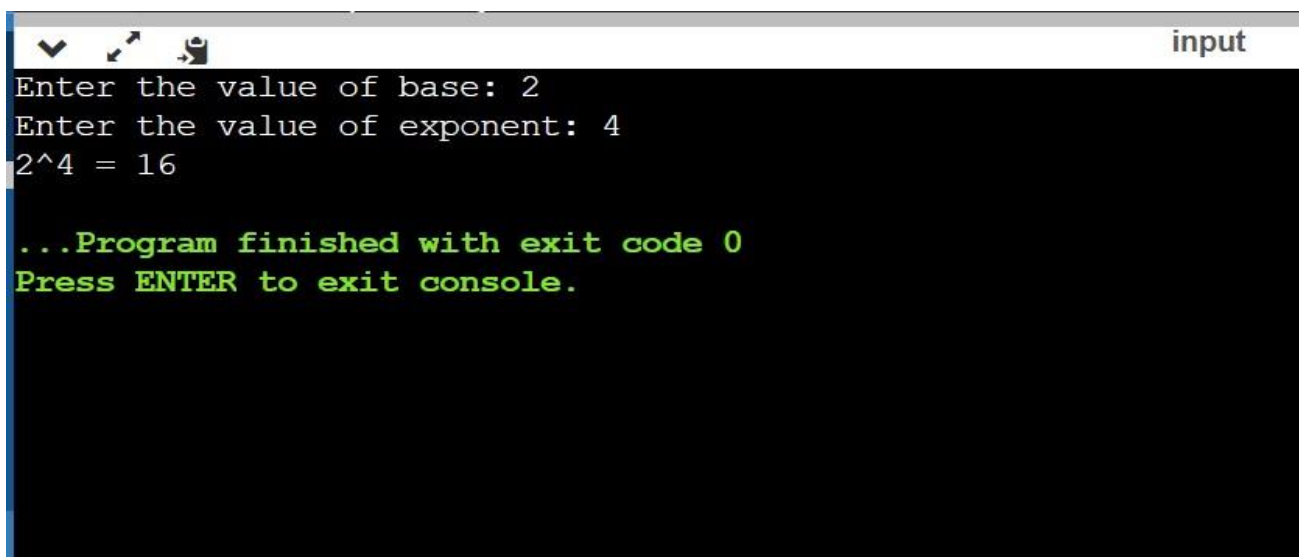
**Space Complexity:**  $O(1)$

## Result/Output/Writing Summary:



```
Enter the value of base: 5
Enter the value of exponent: 2
5^2 = 25

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



```
Enter the value of base: 2
Enter the value of exponent: 4
2^4 = 16

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

## Learning Outcomes:

- Learnt how to implement power function.

- Learnt about recursion.
- Learnt how to analyse time and space complexity