



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment-2

**Student Name:** Tejinderpal Singh

**UID:** 23BCS80343

**Branch:** BE-CSE

**Section/Group:** 23BCS\_KRG\_3A

**Semester:** 5<sup>th</sup>

**Subject Code:** 23CSH-301

**Subject Name:** DAA

**1. Aim:** Code implement power function in O(logn) time complexity.

**2. Objective:** To write a C++ program that finds the power of a number  $x^n$  using a fast method called Exponentiation by Squaring. This method makes the program run faster in  $O(\log n)$  time instead of the slower  $O(n)$  time. The program takes input for base and exponent, calculates the result, and shows the output.

### **3. Procedure:**

1. Start the program and take input for base x and exponent n.
2. Define a function power(x, n) to calculate x raised to the power n.
3. Inside the function, if n is 0, return 1.
4. Initialize a variable output to 1.
5. Use a while loop that continues as long as n is greater than 0.
6. If n is even, square the base x and divide n by 2.
7. If n is odd, multiply output by x and decrease n by 1.
8. Repeat the loop until n becomes 0.
9. Return the final value of output.
10. In the main function, call the power(x, n) function and display the result.

### **4. Code:**

```
#include <iostream>
using namespace std;
int power(int x, int n) {
    if (n == 0) return 1;
```

```
    int output = 1;
    while (n > 0) {
        if (n % 2 == 0) {
            x = x * x;
            n = n / 2;
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        } else {
            output = output * x;
            n--;
        }
    return output;
}

int main() {
    int x, n;
    cout << "Enter base (x): ";
    cin >> x;
    cout << "Enter exponent (n): ";
    cin >> n;
    int result = power(x, n);
    cout << "Result: " << result << endl;
    return 0;
}
```

## 5. Observations:

```
Enter base (x): 2
Enter exponent (n): 2
Result: 4
```

```
Enter base (x): 2
Enter exponent (n): 3
Result: 8
```

## 6. Time Complexity: O(logn)

## 7. Learning Outcome:

- ❖ Learned how to calculate the power of a number using efficient logic (Exponentiation by Squaring).



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

- ❖ Understood how to reduce time complexity from  $O(n)$  to  $O(\log n)$  using a better algorithm.
- ❖ Practiced using loops and conditional statements to solve mathematical problems.
- ❖ Gained experience in writing clean and fast iterative programs in C++.
- ❖ Learned how to take user input, perform calculations, and display the result.