Worksheet-2

Student Name: Arpit Anand
UID: 23BCS12710
Branch: BE-CSE
Section/Group:KRG-3(A)
Semester: 5 **Date of Performance: 29/07/2025**
Subject Name: DAA **Subject Code: 23CSH-301**

1.Aim: Develop a program for implementation of power function and determine that complexity should be O(log n).

2.Objective: To calculate x^n efficiently using recursion with O(logn) time complexity by applying fast exponentiation , including handling of negative powers and base cases.

3.Requirements (Hardware/Software):

Byte XL.

4.Procedure:

```
Public class powerfunc{
   Public static double power (doublex,intn){ if
      (n == 0) return 1;
      if (n < 0) {
         x=1/x; n
         = -n;
      }
      Double half=power(x, n/2); if
      (n % 2 == 0)
         Return half*half;
      else
         Return x*half* half;
```

```
    }
    Public static void main(String[]args){
        System.out.println(power(2, 10));
        System.out.println(power(5,0));
        System.out.println(power(2,-3));
        System.out.println(power(3,5));
        System.out.println(power(10,2));
    }
}
```

Algorithm:

1. Start
2. Input:basex,exponent n.
3. If n==0,return 1.
4. If n <0, set x=1/x and n = -n.
5. Compute half=power(x, n/2) recursively.
6. If n is even, return half * half.
7. Else(n is odd),return x*half* half.
8. End

Time Complexity: O(logn)

Space complexity: O(logn)

Output:

Learning Outcomes:

1. Learned how to calculate powers using recursion instead of repeated multiplication.
2. Understood how dividing the problem into halves makes the program faster (O(log n)).
3. Learned to handle negative powers and base cases correctly in code.