LABORATORY REPORT

Algorithm Laboratory (CS-39001)

B.Tech Program in ECSc

Submitted By

Name:- SANNIDHI DEB

Roll No: 2330044

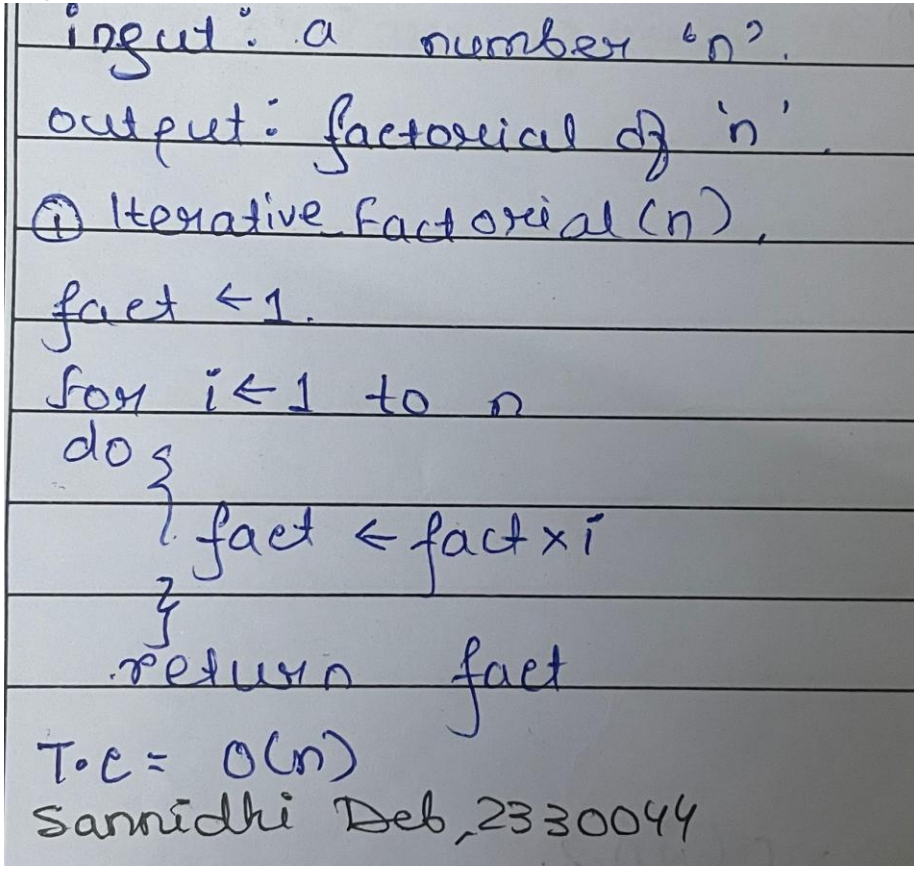
Kalinga Institute of Industrial Technology (Deemed to be University) Bhubaneswar, India

Autumn, 2025

Table of Contents

Experiment Number	2.1
Experiment Title	Aim of the program: Design an algorithm to find the factorial of a positive integer n. Analyze the time complexity of your algorithm. Implement a C program to find the factorial of a positive integer n.
Date of Experiment	07/08/2025
Date of Submission	20/08/2025

1. **Algorithm:-**

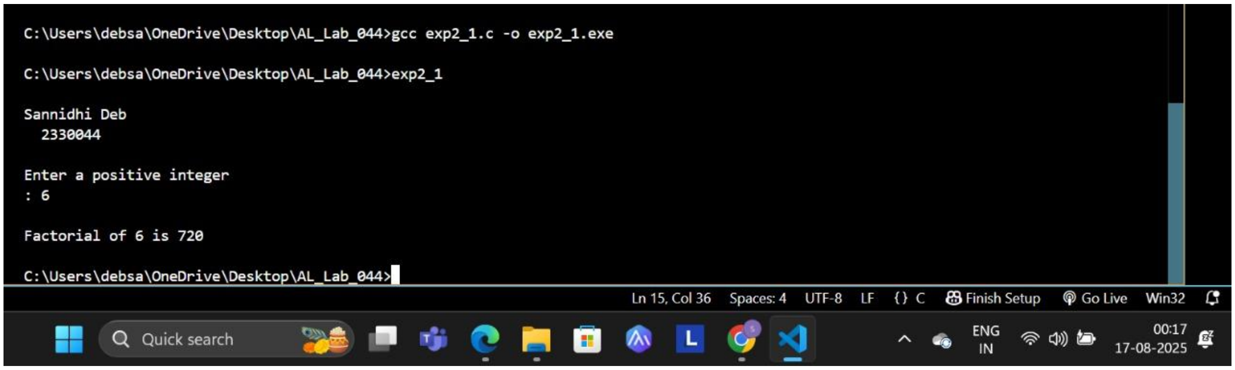


```
input: a number 'n'.
output: factorial of 'n'.
① Iterative_Factorial(n),
fact ← 1.
for i ← 1 to n
 do {
     fact ← fact × i
    }
  return fact
T.C = O(n)
Sannidhi Deb, 2330044
```

2. Code:-

```
#include <stdio.h>

int factorial(int n) {
   int result = 1;
   for (int i = 2; i <= n; i++)
      result *= i;
   return result;
}

int main() {
   int n;
   printf("\nSannidhi Deb\n  2330044\n\n");
   printf("Enter a positive integer\n: ");
   scanf("%d", &n);
   printf("\nFactorial of %d is %d\n", n, factorial(n));
   return 0;
}
```

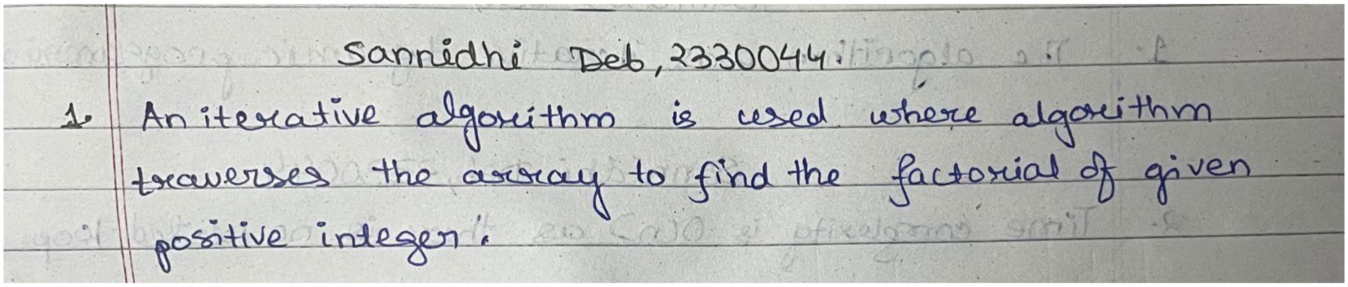
3. Results/Output:- Entire Screen Shot including Date & Time:-



```
C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>gcc exp2_1.c -o exp2_1.exe

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>exp2_1

Sannidhi Deb
  2330044

Enter a positive integer
: 6

Factorial of 6 is 720

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>
```

4. Remarks:-

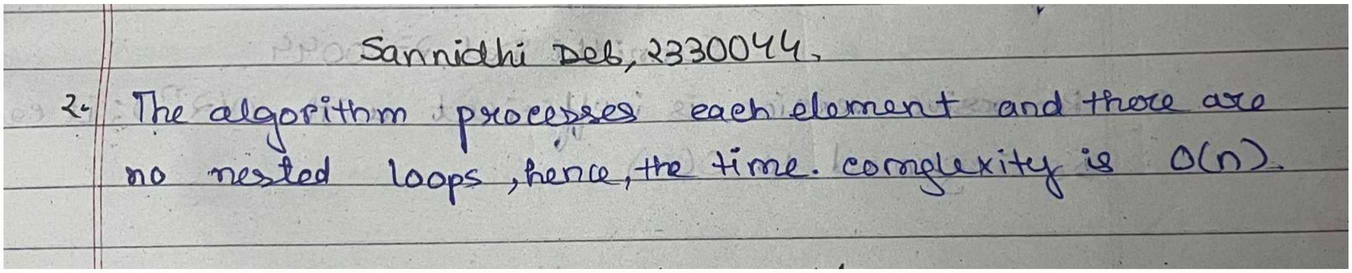
1. What type of algorithm is used?



Sannidhi Deb, 2330044.

1. An iterative algorithm is used where algorithm traverses the array to find the factorial of given positive integer.

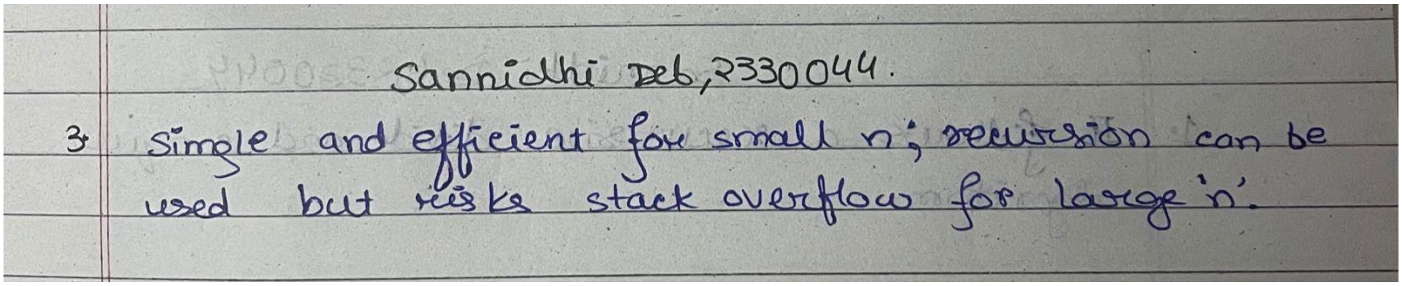
2. Analyze the complexity of your algorithm.



Sannidhi Deb, 2330044.

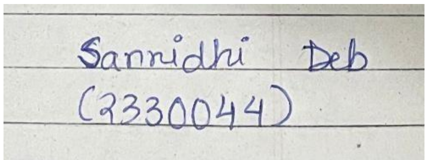
2. The algorithm processes each element and there are no nested loops, hence, the time. complexity is O(n).

3.Any other observations?



Sannidhi Deb, 2330044.

3. Simple and efficient for small n; recursion can be used but risks stack overflow for large 'n'.



Sannidhi Deb
(2330044)

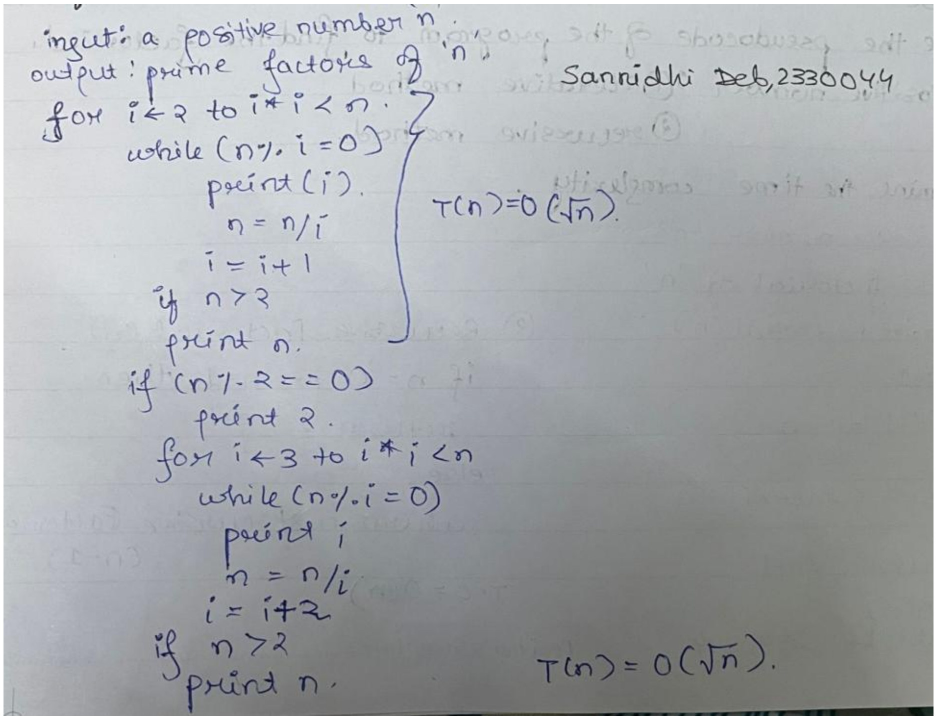
Signature of the FIC

Sannidhi Deb

(Name of the FIC)

Experiment Number	2.2
Experiment Title	Design an algorithm to find the prime factorization of a positive integer n. Analyze the time complexity of your algorithm. Implement a C program to find the prime factorization of a positive integer n.
Date of Experiment	07/08/2025
Date of Submission	20/08/2025

1. Algorithm:-



Sannidhi Deb,2330044

```
input: a positive number n.
output: prime factors of 'n'.
for i←2 to i*i < n.
    while (n%. i = 0)
        print (i).
        n = n/i
    i = i+1
  if n>2
  print n.
```

$T(n) = O(\sqrt{n})$.

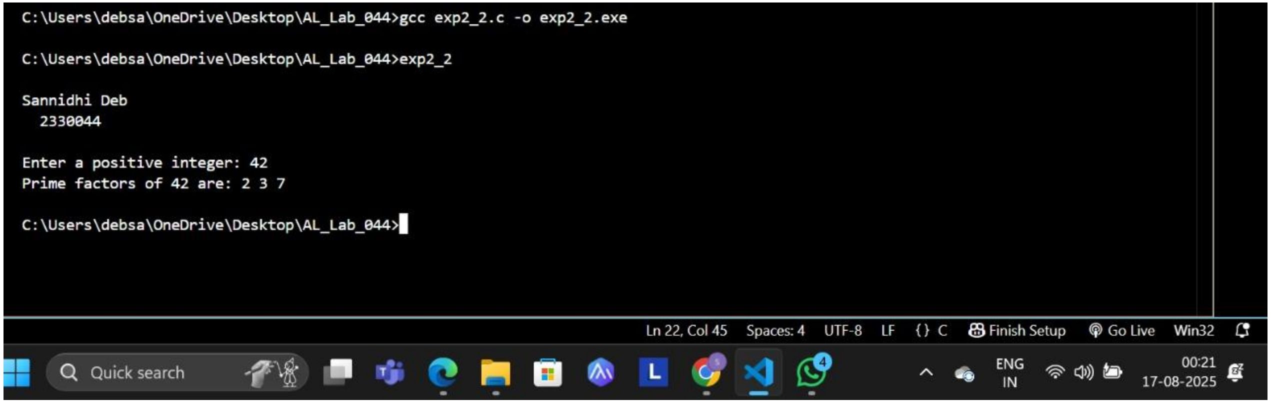
```
if (n%.2 == 0)
    print 2.
for i←3 to i*i <n
    while (n%.i = 0)
        print i
        n = n/i
    i = i+2
if n>2
print n.
```

$T(n) = O(\sqrt{n})$.

2. Code:-

```
#include <stdio.h>

void primeFactors(int n) {
    while (n % 2 == 0) {
        printf("2 ");
        n /= 2;
    }

    for (int i = 3; i * i <= n; i += 2) {
        while (n % i == 0) {
            printf("%d ", i);
            n /= i;
        }
    }

    if (n > 2)
        printf("%d", n);
}

int main() {
    int n;
    printf("\nSannidhi Deb\n  2330044\n\n");
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Prime factors of %d are: ", n);
    primeFactors(n);
    printf("\n");
    return 0;
}
```

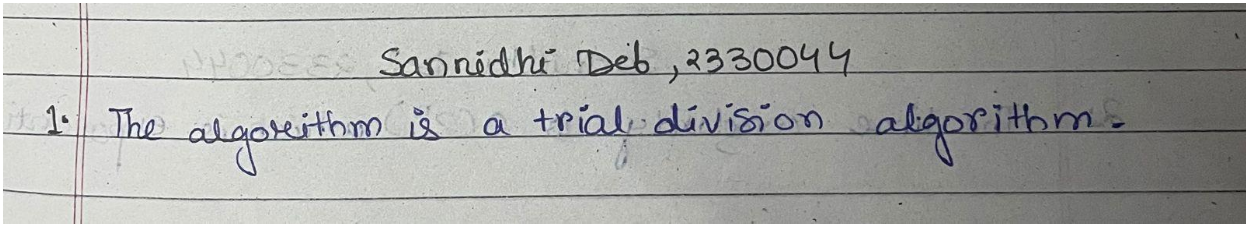
3. Results/Output:- Entire Screen Shot including Date & Time:-



```
C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>gcc exp2_2.c -o exp2_2.exe

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>exp2_2

Sannidhi Deb
  2330044

Enter a positive integer: 42
Prime factors of 42 are: 2 3 7

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>
```

4. Remarks:-

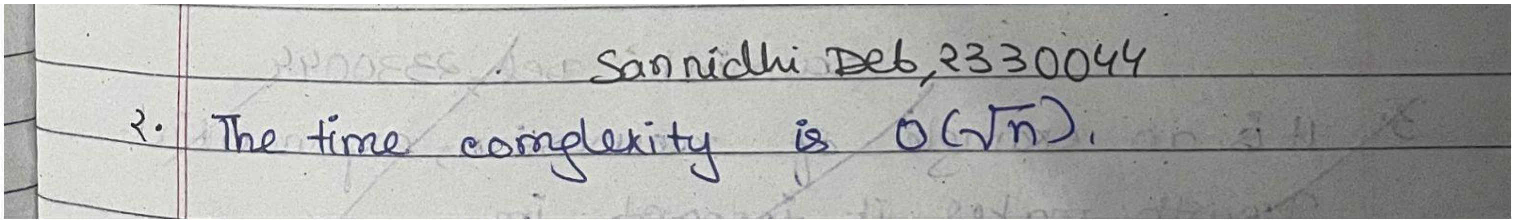
1. What type of algorithm is used?



Sannidhi Deb ,2330044

1. The algorithm is a trial division algorithm.

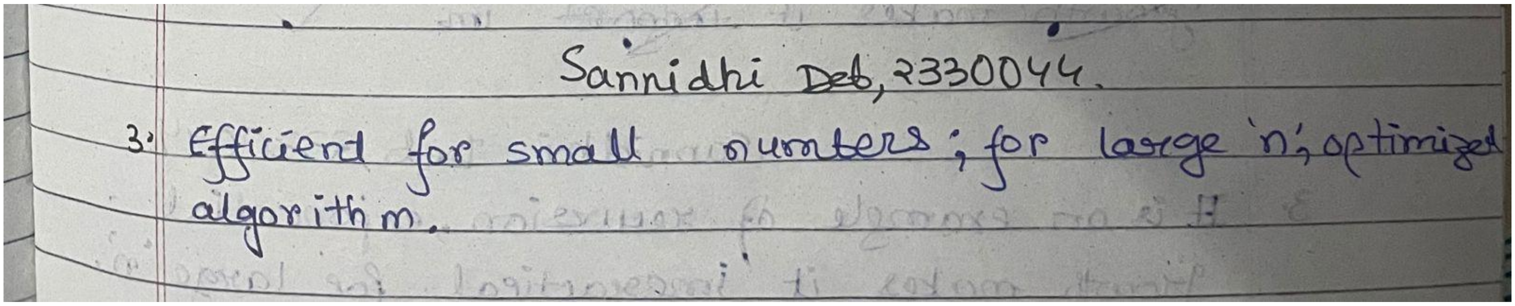
2. Analyze the complexity of your algorithm.



Sannidhi Deb, 2330044

2. The time complexity is $O(\sqrt{n})$.

3.Any other observations?



Sannidhi Deb, 2330044.

3. Efficient for small numbers; for large 'n', optimized algorithm.

Sannidhi Deb
(2330044)

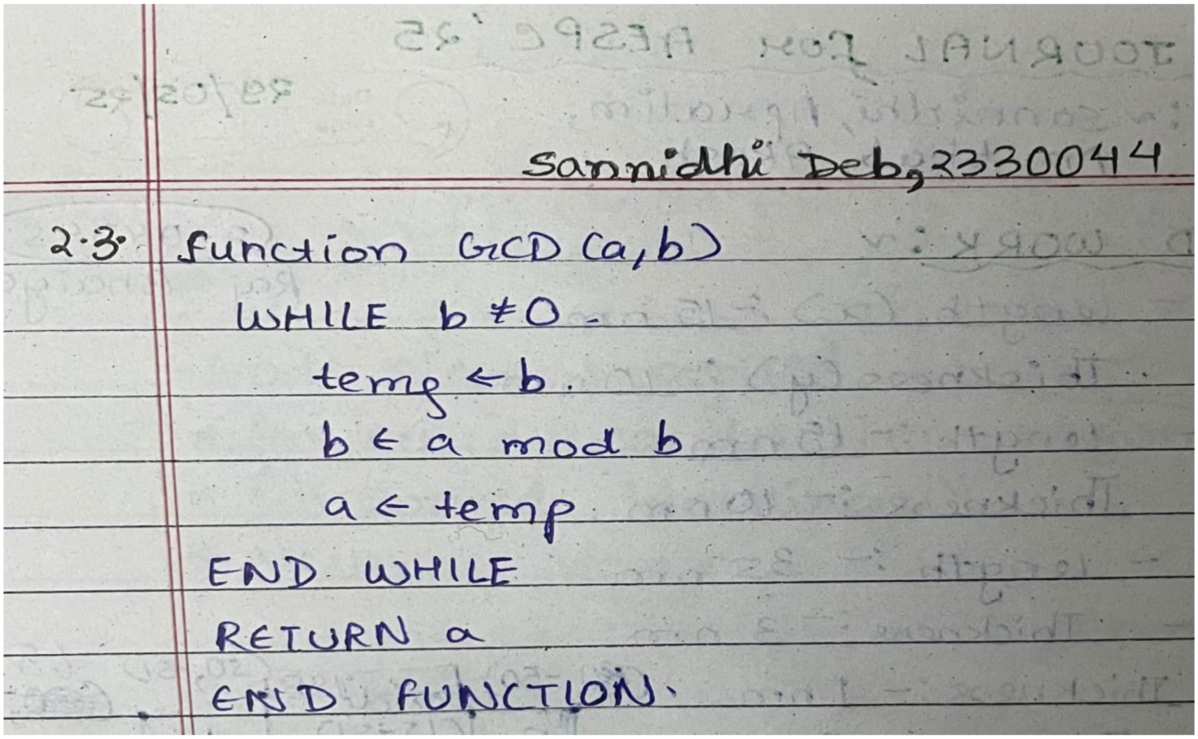
Signature of the FIC

Sannidhi Deb

(Name of the FIC)

Experiment Number	2.3
Experiment Title	Design an algorithm to find the greatest common divisor (GCD) of two integers. Analyze the time complexity of your algorithm. Implement a C program to find the GCD of two integers.
Date of Experiment	07/08/2025
Date of Submission	20/08/2025

1. Algorithm:-



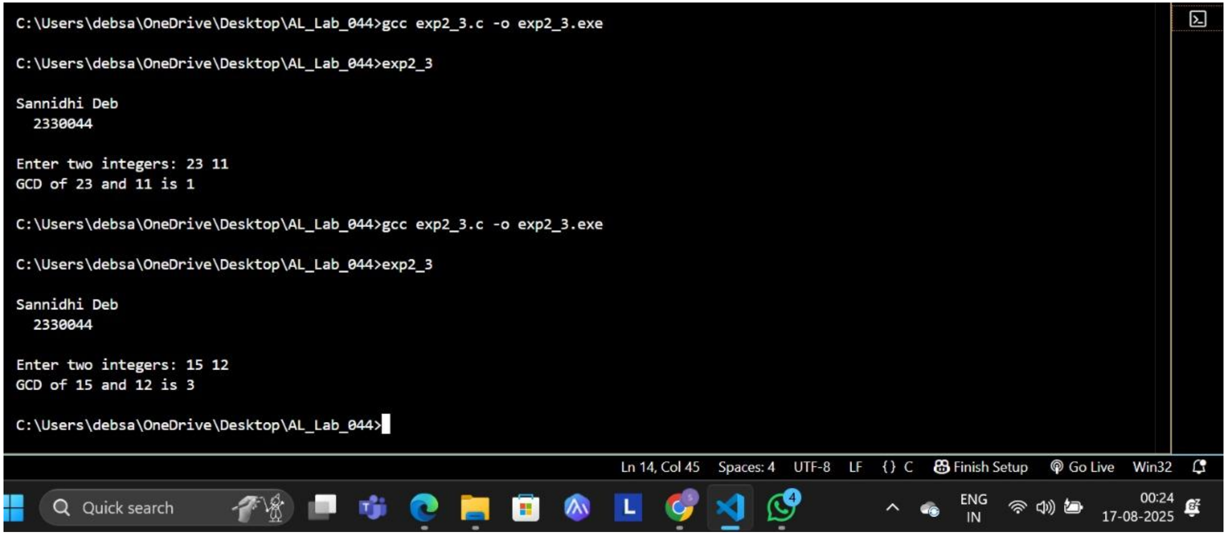
Sannidhi Deb, 2330044

```
2.3. function GCD (a,b)
       WHILE b ≠ 0 .
          temp ← b.
          b ← a mod b
          a ← temp
       END WHILE
       RETURN a
     END FUNCTION.
```

2. Code:-

```
#include <stdio.h>

int gcd(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

int main() {
    int a, b;
    printf("\nSannidhi Deb\n  2330044\n\n");
    printf("Enter two integers: ");
    scanf("%d %d", &a, &b);
    printf("GCD of %d and %d is %d\n", a, b, gcd(a, b));
    return 0;
}
```

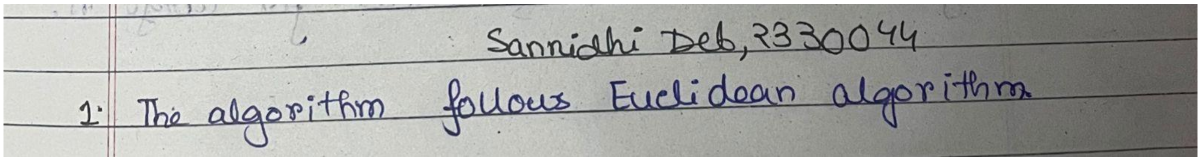
3. Results/Output:- Entire Screen Shot including Date & Time:-



```
C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>gcc exp2_3.c -o exp2_3.exe

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>exp2_3

Sannidhi Deb
  2330044

Enter two integers: 23 11
GCD of 23 and 11 is 1

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>gcc exp2_3.c -o exp2_3.exe

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>exp2_3

Sannidhi Deb
  2330044

Enter two integers: 15 12
GCD of 15 and 12 is 3

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>
```

4. Remarks:-

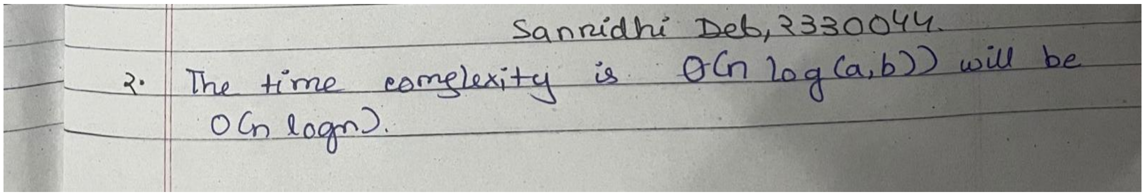
1. What type of algorithm is used?



Sannidhi Deb, 2330044

1. The algorithm follows Euclidean algorithm

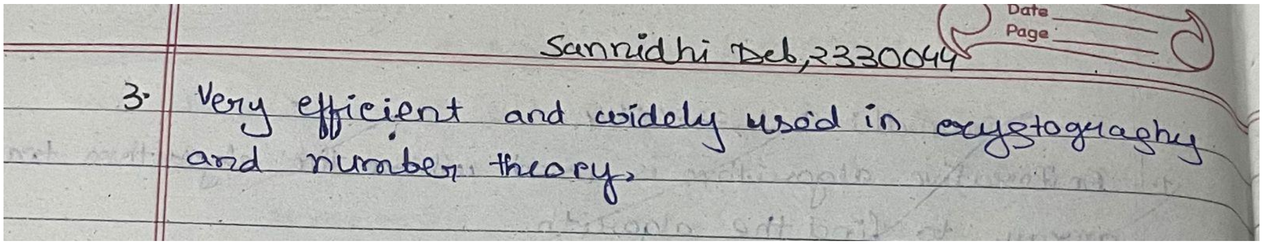
2. Analyze the complexity of your algorithm.



Sannidhi Deb, 2330044.

2. The time complexity is $\Theta(n \log(a,b))$ will be $O(n \log n)$.

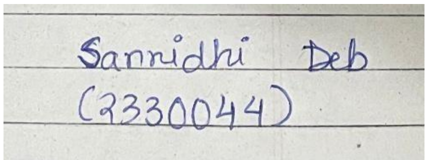
3.Any other observations?



Date
Page

Sannidhi Deb, 2330044

3. Very efficient and widely used in crystography and number theory.



Sannidhi Deb
(2330044)

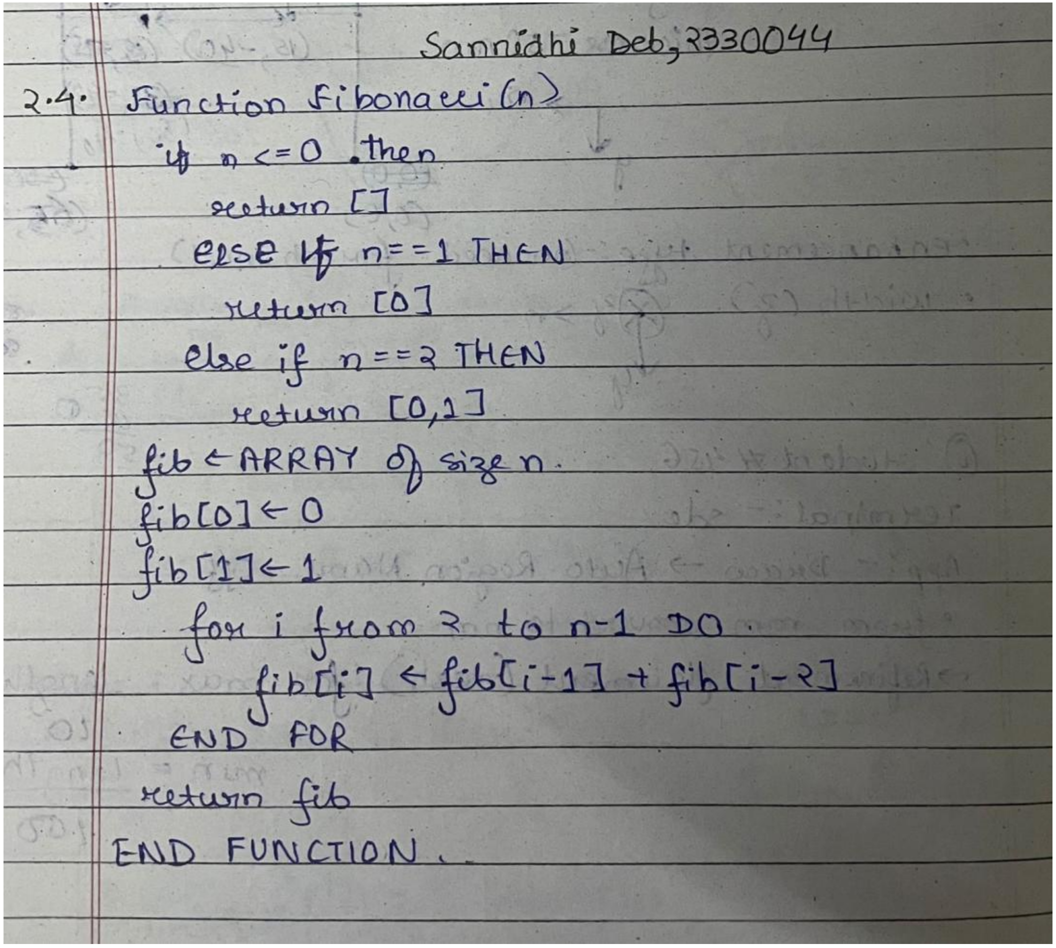
Signature of the FIC

Sannidhi Deb

(Name of the FIC)

Experiment Number	2.4
Experiment Title	Design an algorithm to find the Fibonacci series of nth number. Analyze the time complexity of your algorithm. Implement a C program to find the Fibonacci series of nth number.
Date of Experiment	07/08/2025
Date of Submission	20/08/2025

1. Algorithm:-



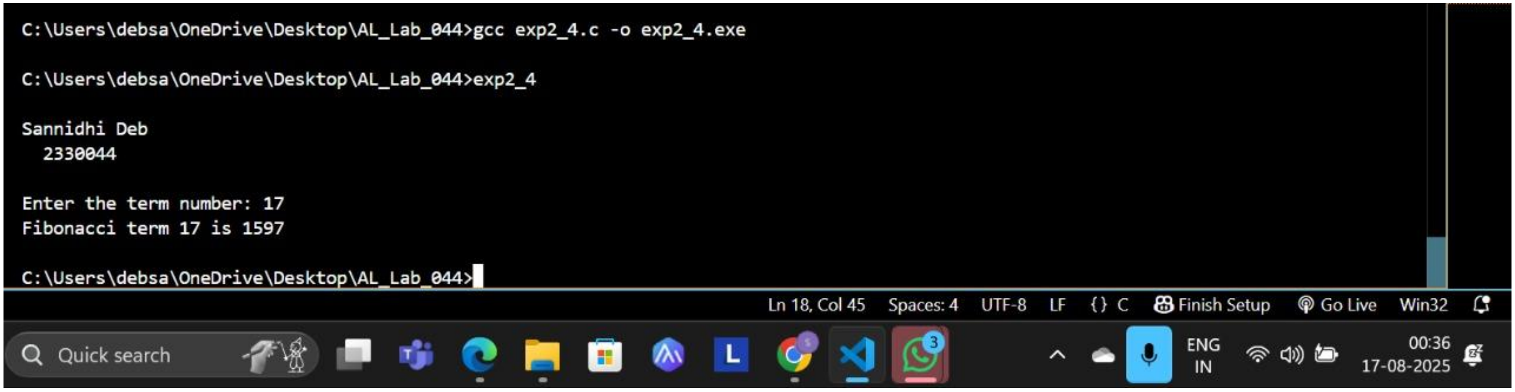
Sannidhi Deb, 2330044

```
2.4· Function fibonacci(n)
      if n <= 0 .then
         return []
       else if n==1 THEN
          return [0]
       else if n==2 THEN
          return [0,1]
     fib ← ARRAY of size n.
     fib[0] ← 0
     fib[1] ← 1
       for i from 2 to n-1 DO
           fib[i] ← fib[i-1] + fib[i-2]
       END FOR
     return fib
    END FUNCTION.
```

2. Code:-

```
#include <stdio.h>

int fibonacci(int n) {
    if (n == 0) return 0;
    if (n == 1) return 1;

    int a = 0, b = 1, c;
    for (int i = 2; i <= n; i++) {
        c = a + b;
        a = b;
        b = c;
    }
    return b;
}

int main() {
    int n;
    printf("\nSannidhi Deb\n  2330044\n\n");
    printf("Enter the term number: ");
    scanf("%d", &n);
    printf("Fibonacci term %d is %d\n", n, fibonacci(n));
    return 0;
}
```

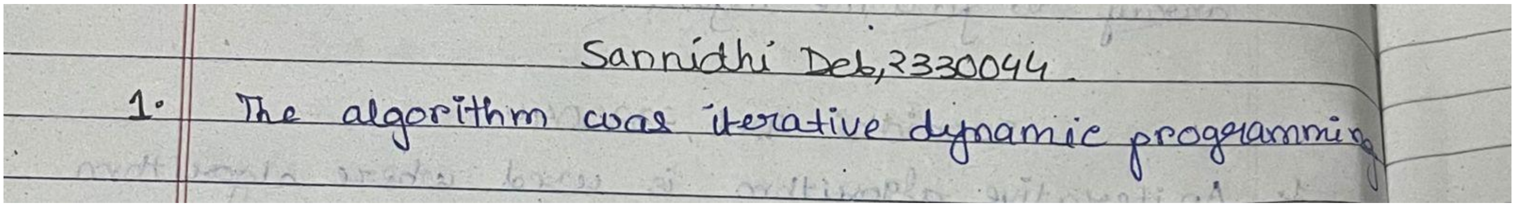
3. Results/Output:- Entire Screen Shot including Date & Time:-



```
C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>gcc exp2_4.c -o exp2_4.exe

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>exp2_4

Sannidhi Deb
  2330044

Enter the term number: 17
Fibonacci term 17 is 1597

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>
```

4. Remarks:-

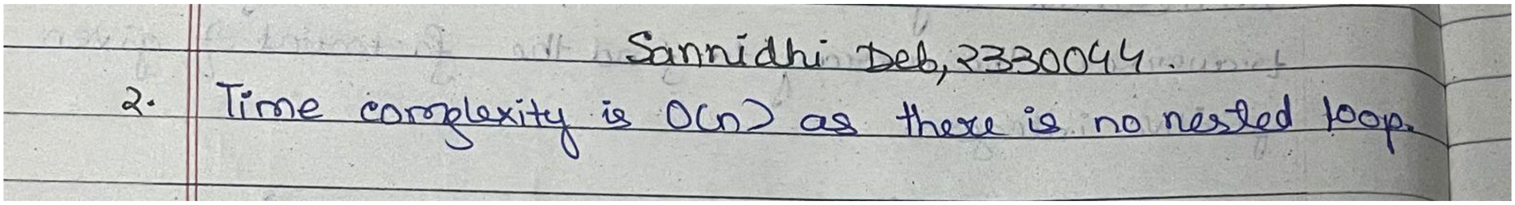
1. What type of algorithm is used?



Sannidhi Deb,2330044.

1. The algorithm was iterative dynamic programming

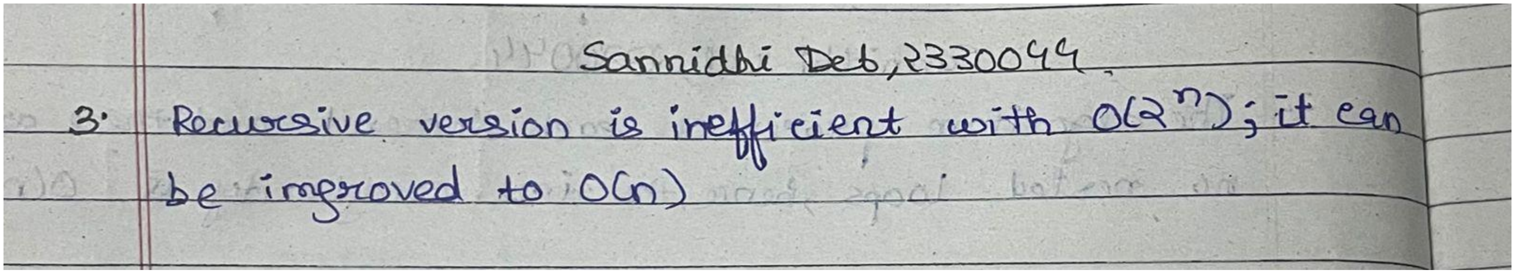
2. Analyze the complexity of your algorithm.



Sannidhi Deb,2330044.

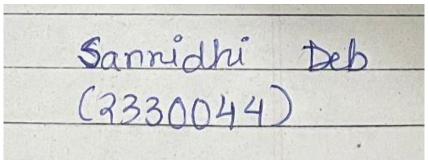
2. Time complexity is O(n) as there is no nested loop.

3.Any other observations?



Sannidhi Deb,2330044.

3. Recursive version is inefficient with $O(2^n)$; it can be improved to O(n)



Sannidhi Deb
(2330044)

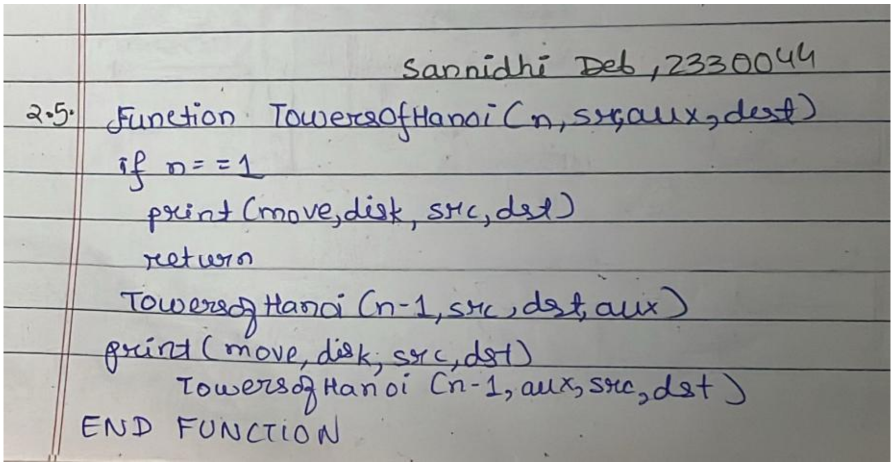
Signature of the FIC

Sannidhi Deb

(Name of the FIC)

Experiment Number	2.5
Experiment Title	The Towers of Hanoi is a classic mathematical puzzle where you are given three rods (let's call them A, B, and C) and a number of disks of different sizes that can slide onto any rod. Initial setup i) All disks start on Rod A, stacked in decreasing order of size (largest at bottom, smallest at top) ii) Rods B and C are initially empty. Objective Move all disks from Rod A to Rod C. Rules i)You can only move one disk per operation. ii)You can only move the top disk from any rod. iii)A larger disk can never be placed on top of a smaller disk. iv)A disk can be moved from any rod to any other rod as long as it does not violate the size constraint. Design an algorithm that solves the Towers of Hanoi puzzle for n disks. Analyze the time complexity of your algorithm. Also write a C program to implement the same
Date of Experiment	07/08/2025
Date of Submission	20/08/2025

1. Algorithm:-



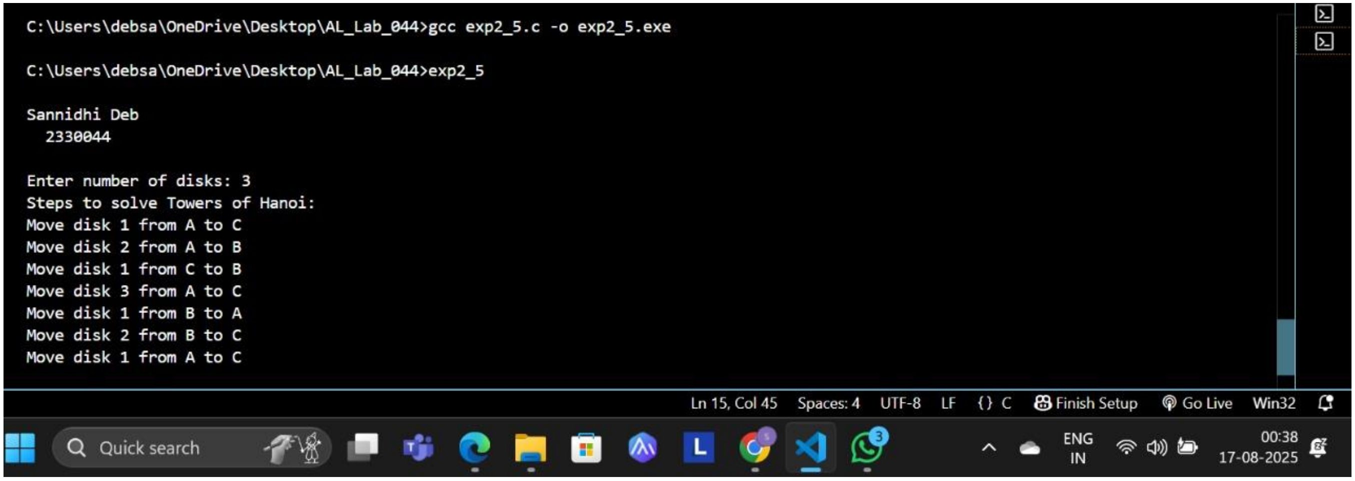
Sannidhi Deb, 2330044

```
2.5. Function TowersofHanoi(n, src,aux, dest)
      if n==1
        print(move,disk, src, dst)
        return
      TowersofHanoi(n-1, src, dst, aux)
     print(move, disk, src, dst)
         TowersofHanoi (n-1, aux, src, dst)
    END FUNCTION
```

2. Code :-

```
#include <stdio.h>

void hanoi(int n, char source, char target, char auxiliary) {
   if (n == 1) {
      printf("Move disk 1 from %c to %c\n", source, target);
      return;
   }
   hanoi(n - 1, source, auxiliary, target);
   printf("Move disk %d from %c to %c\n", n, source, target);
   hanoi(n - 1, auxiliary, target, source);
}

int main() {
   int n;
   printf("\nSannidhi Deb\n  2330044\n\n");
   printf("Enter number of disks: ");
   scanf("%d", &n);
   printf("Steps to solve Towers of Hanoi:\n");
   hanoi(n, 'A', 'C', 'B');
   return 0;
}
```

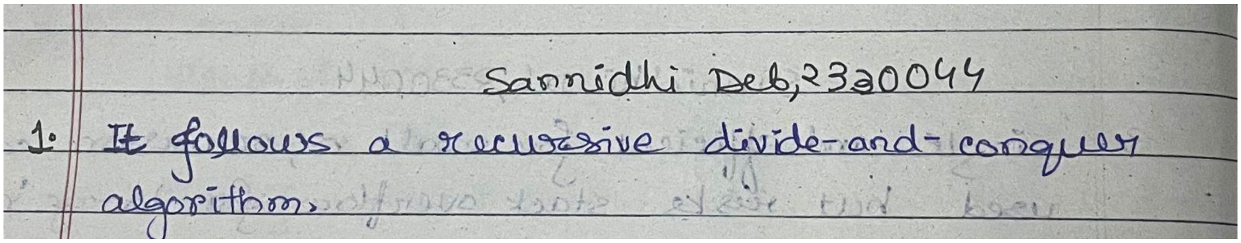
3. Results/Output:- Entire Screen Shot including Date & Time:-



```
C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>gcc exp2_5.c -o exp2_5.exe

C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>exp2_5

Sannidhi Deb
  2330044

Enter number of disks: 3
Steps to solve Towers of Hanoi:
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
```

4. Remarks:-

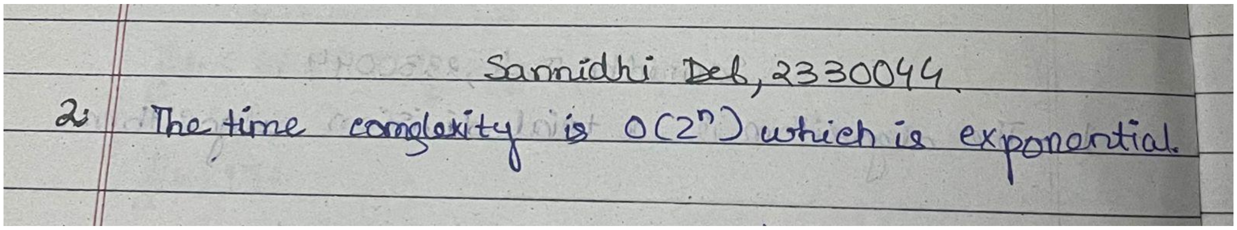
1.What type of algorithm is used?



Sannidhi Deb,2330044

1. It follows a recursive divide-and- conquer algorithm.

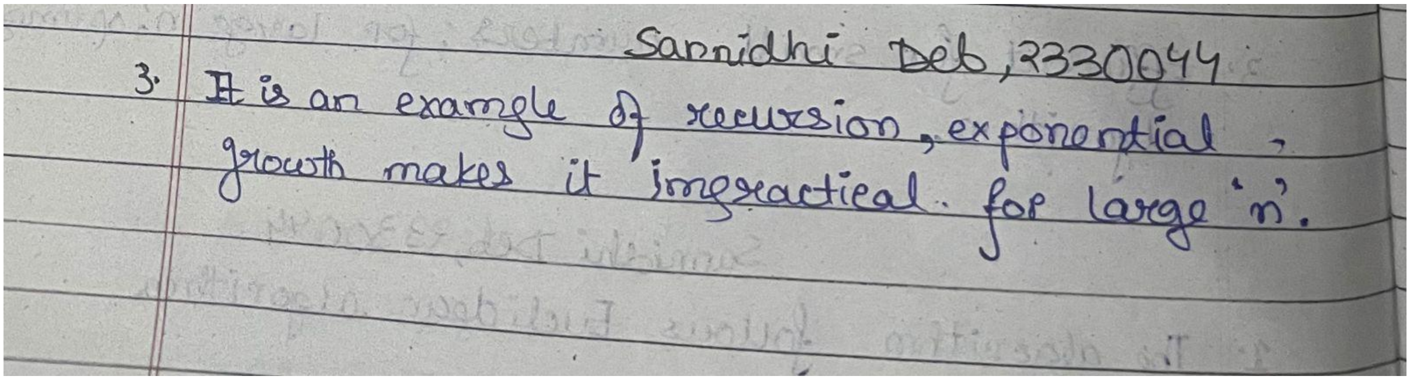
2.Analyze the complexity of your algorithm.



Sannidhi Deb, 2330044

2. The time complexity is $O(2^n)$ which is exponential.

3.Any other observations?



Sannidhi Deb, 2330044

3. It is an example of recursion, exponential , growth makes it impractical. for large 'n'.

5. Conclusion:-

In this lab session, we revised essential operations in C programming. We implemented programs to find the factorial, get prime factors, find GCD of integers, Fibonacci terms and also perform a problem on the TowersofHanoi. These exercises reinforced key data structure concepts such as array traversal, in-place updates, and efficient element manipulation — all fundamental for building more complex algorithms.

Sannidhi Deb
(2330044)

Signature of the FIC

Sannidhi Deb

(Name of the FIC)