B5

## Task-5

## Basic Number Theory 1

TICTAC TOE

Aim :- To write and execute the program for given Scenario based on Basic Number Theory-1

Algorithm

Read the integer n.

. Calculate the minimum value of the sum as $(n*(n-1)(2n-1))/6$ modulo 109+7.

. Calculate the maximum value of the sum as $((n*(n-1))/2*n-((n(n-1)/2)*(2*n-1))/3$ modulo 109+7.

Print the minimum and maximum values of the sum as output.

End the program.

# Task 5: Basic Number Theory 1

**1.Program:**

```
#include <stdio.h>
#include <math.h>

#define MOD 1000000007
int main() { int
  t;
  scanf("%d",      &t);
  while  (t--)  {  long
  long n; scanf("%lld",
  &n);
    long long min_sum = ((n - 1) * (n - 1)*n /4) % MOD; // Minimum sum long long
    max_sum  =  ((n  *  (n  +  1))  *  (2  *( n  +  1))/(12)) % MOD; // Maximum sum
    printf("%lld %lld\n", min_sum, max_sum);
 }  return
 0;
}
```

25

## Problem

Given two integers and a recursive technique to find their GCD is the Euclidean Algorithm. The algorithm states that for Compating the GCD of the two positive integers and it and are equal. Otherwise if, there are a few optimizations that an be made to the above logic to arrive at a more efficient mplementation.

## lgorithm

Read the two positive integers a and b.
If b is zero, return a as the GCD.
Otherwise, recursively call the function with arguments b an
the remainder of a divided by b.
Return the result the recursive call as the GCD.

**2.Program:**

```
#include  <stdio.h>
int gcd(int a, int b) {
if (b == 0) { return
a;
   }
   return gcd(b, a % b);
} int main()
{ int a, b;
   scanf("%d %d", &a, &b);
int   result   =   gcd(a,   b);
printf("%d\n",         result);
return 0; }
```

Output

Sample Input

1 5

Sample Output

1

| | |
|---|---|
| … TECH · CSE | |
| …O. | 8 |
| …FORMANCE (5) | 5 |
| …ULT AND ANALYSIS (3) | 3 |
| …A VOCE (3) | 2 |
| RECORD (4) | 4 |
| TOTAL (15) | 14 |
| … WITH DATE | |

sult :-

Thus, the program is executed and verified successfull