**Theory:**

Totient function also known as Euler’s Totient function can be defined as the number of positive integer less than n, which are relatively prime to n. It is denoted by Φ(n).

Example: Φ(10)=?

Here, n = 10

Numbers less than 10 are: {1,2,3,4,5,6,7,8,9}

Numbers relatively prime to 10 are: {1,3,7,9}

∴ Φ(10) = 4

If n is the prime number then Φ(n) = n-1

Example: Φ(7) = ?

Φ(7) = 7-1

= 6

Let p and q be the two prime numbers such that p ≠ q and n = p\*q, then Φ(n) = (p-1)(q-1) Example: Φ(15) = ?

Here,

15 = 3 \* 5

p = 3 and q = 5 then

Φ(15) = (3-1)(5-1)

= 2 \* 4

= 8

**Programming Language: C**

**IDE: VS-Code**

**Code :**

#include <stdio.h>

// gcd calculation

int gcd(int a, int b)

{

if (a == 0)

return b;

return gcd(b % a, a);

}

// Totient Function

int phi(unsigned int n)

{

int i;

unsigned int result = 1;

for (i = 2; i < n; i++)

if (gcd(i, n) == 1)

result++;

return result;

}

int main()

{

int n;

printf("Enter the number:");

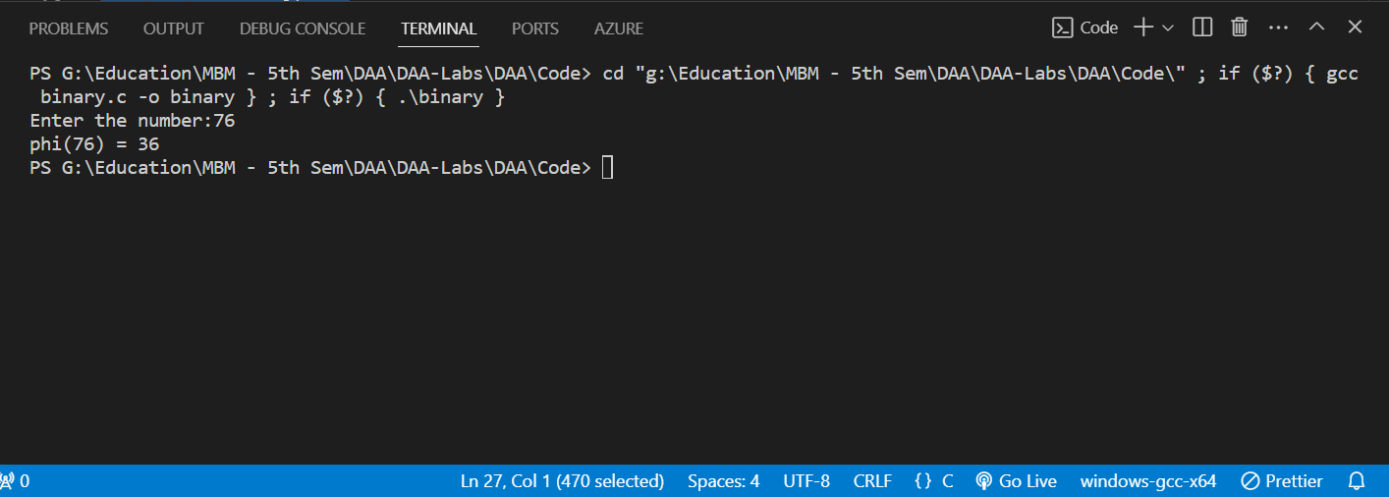
scanf("%d", &n);

printf("phi(%d) = %d\n", n, phi(n));

return 0;

}

**Output**

****