**SOL 1:**

**#include <stdio.h>**

**#include <math.h>**

**int isArmstrong(int num);**

**int isPerfect(int num);**

**int main()**

**{**

**int num;**

**printf("Enter any number: ");**

**scanf("%d", &num);**

**if(isArmstrong(num))**

**{**

**printf("%d is Armstrong number.\n", num);**

**}**

**else**

**{**

**printf("%d is not Armstrong number.\n", num);**

**}**

**if(isPerfect(num))**

**{**

**printf("%d is Perfect number.\n", num);**

**}**

**else**

**{**

**printf("%d is not Perfect number.\n", num);**

**}**

**return 0;**

**}**

**int isArmstrong(int num)**

**{**

**int lastDigit, sum, originalNum, digits;**

**sum = 0;**

**originalNum = num;**

**digits = (int) log10(num) + 1;**

**while(num > 0)**

**{**

**lastDigit = num % 10;**

**sum = sum + round(pow(lastDigit, digits));**

**num = num / 10;**

**}**

**return (originalNum == sum);**

**}**

**int isPerfect(int num)**

**{**

**int i, sum, n;**

**sum = 0;**

**n = num;**

**for(i=1; i<n; i++)**

**{**

**if(n%i == 0)**

**{**

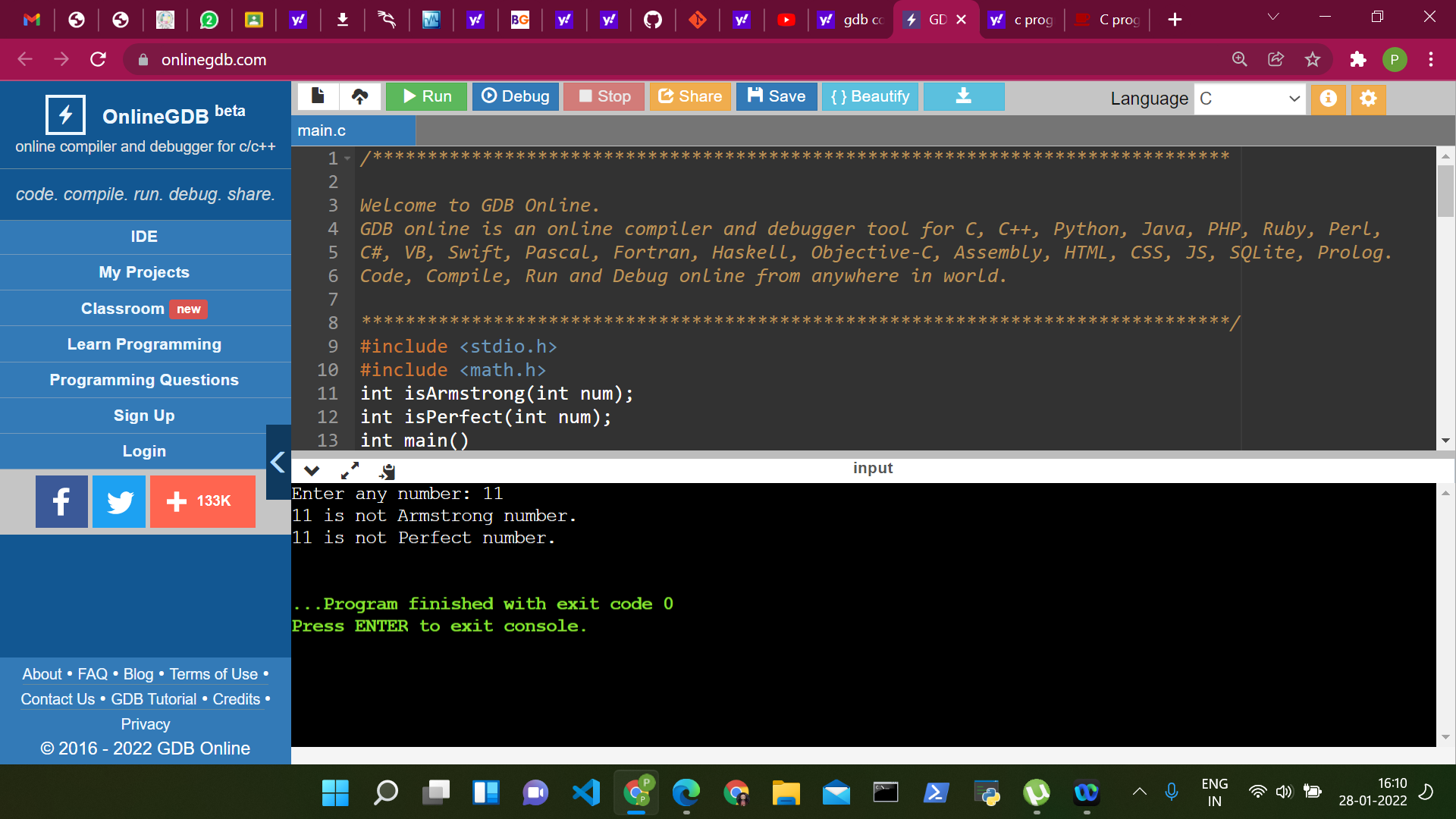
**sum += i;**

**}**

**}**

**return (num == sum);**

**}**



**SOL 2:**

**#include<stdio.h>**

**int sumOfRange(int);**

**int main()**

**{**

**int n1;**

**int sum;**

**printf("Enter Last Term: ");**

**scanf("%d", &n1);**

**sum = sumOfRange(n1);**

**printf("\n The sum of numbers from 1 to %d : %d\n\n", n1, sum);**

**return (0);**

**}**

**int sumOfRange(int n1)**

**{**

**int res;**

**if (n1 == 1)**

**{**

**return (1);**

**}**

**else**

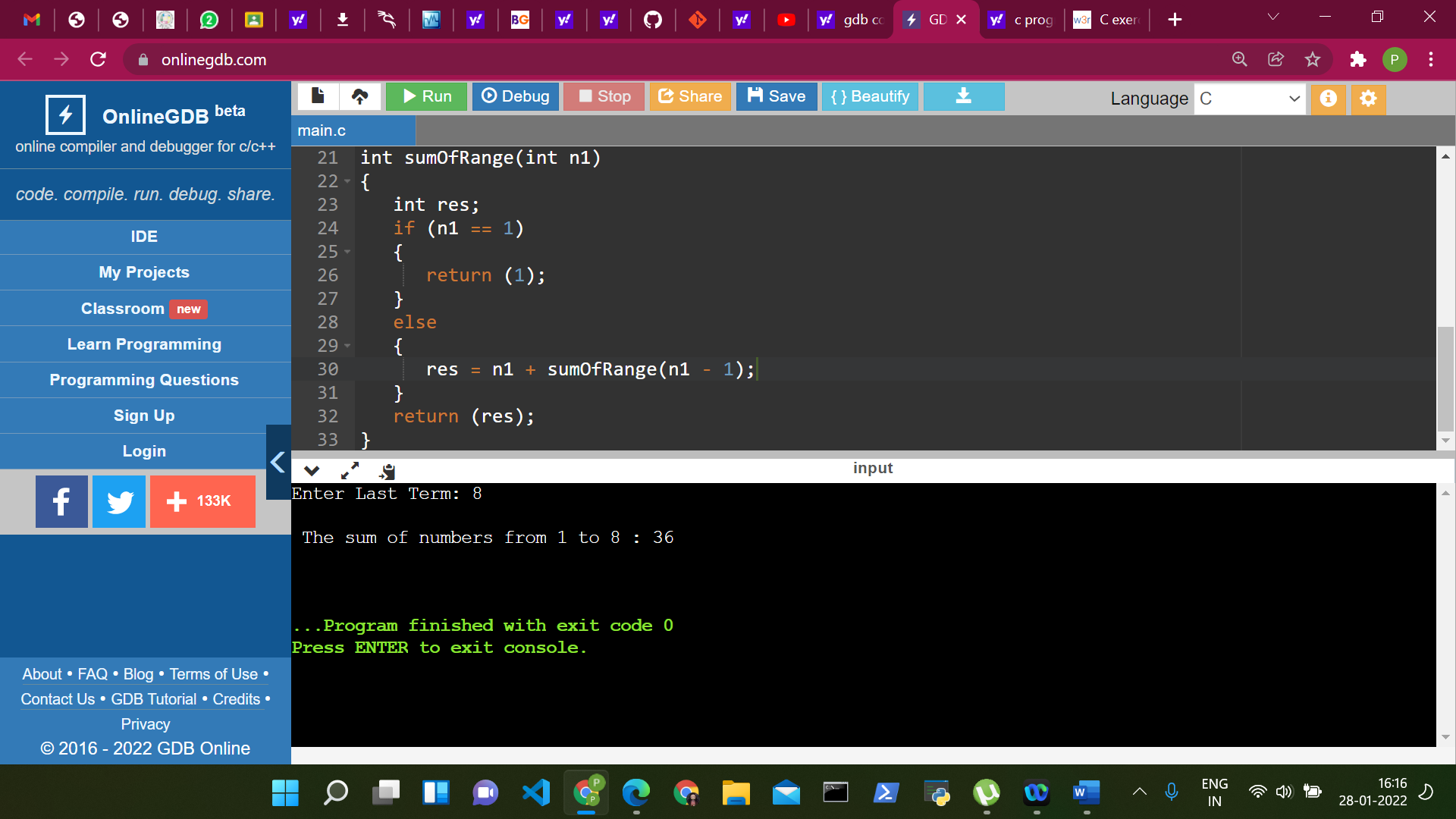
**{**

**res = n1 + sumOfRange(n1 - 1);**

**}**

**return (res);**

**}**



**SOL 3:**

**#include<stdio.h>**

**int count\_dig(int number);**

**int main()**

**{**

**int number,count;**

**printf(" Input a number : ");**

**scanf("%d",&number);**

**count = count\_dig(number);**

**printf(" The number of digits in the number is : %d \n\n",count);**

**return 0;**

**}**

**int count\_dig(int number)**

**{**

**static int count=0;**

**if(number!=0)**

**{**

**count++;**

**count\_dig(number/10);**

**}**

**return count;**

**}**

