C ++ ASSIGNMENT 2 (FUNCTIONS)

1. WAP for printing all natural numbers till 20.

#include <iostream>

using namespace std;

void num(int i) // creating a function

{

for (int j = 1; j <= i; ++j) {

cout <<j<< " ";

}

cout << endl;

}

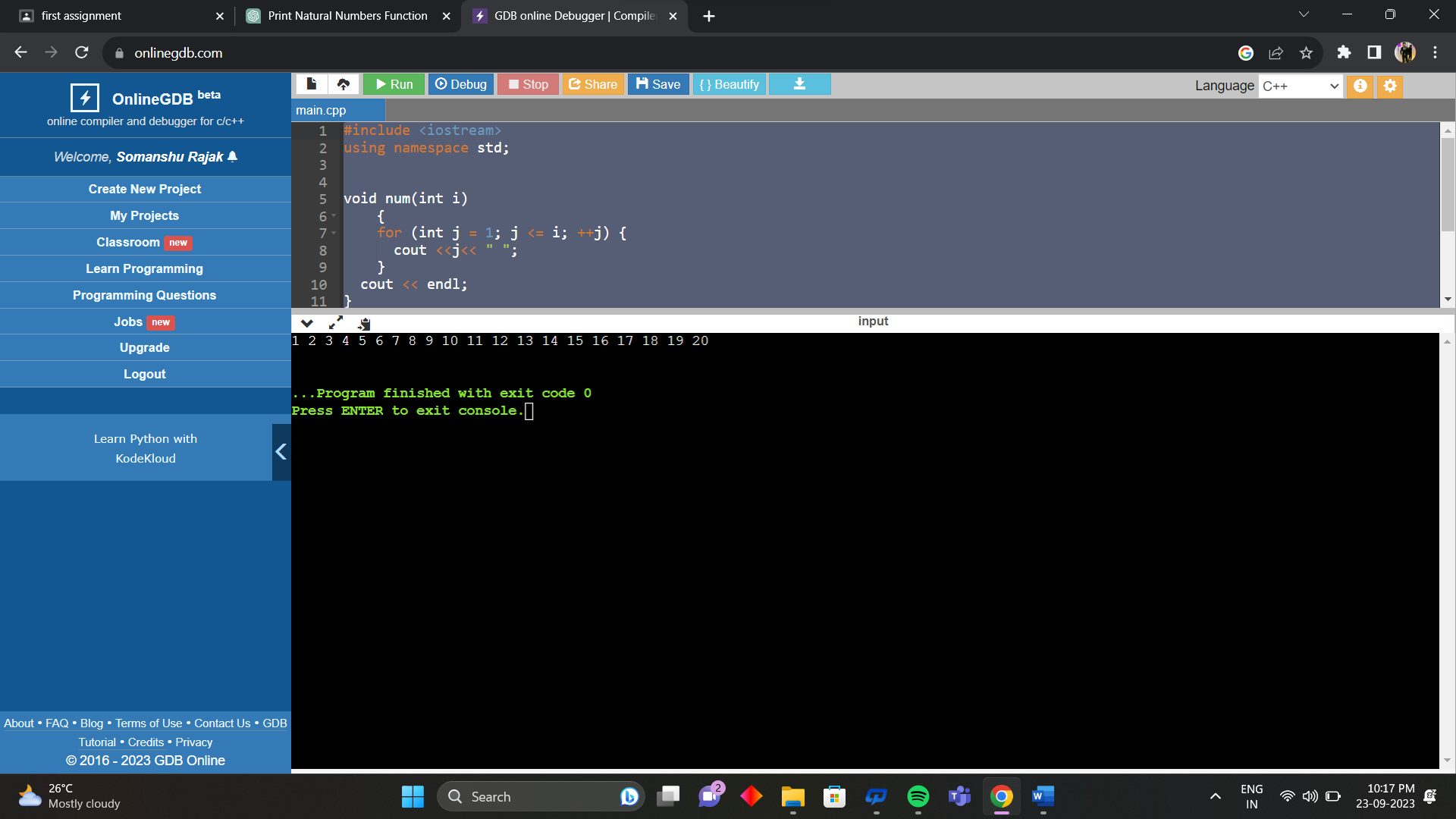
int main() {

int n = 20;

num(n); // calling function

return 0;

}



2. WAP for printing all natural numbers in reverse order starting from 20.

#include <iostream>

using namespace std;

void num(int i) // function name num//

{

for (int j = 20; i<=j; --j)

{

cout <<j<< " ";

}

cout << endl;

}

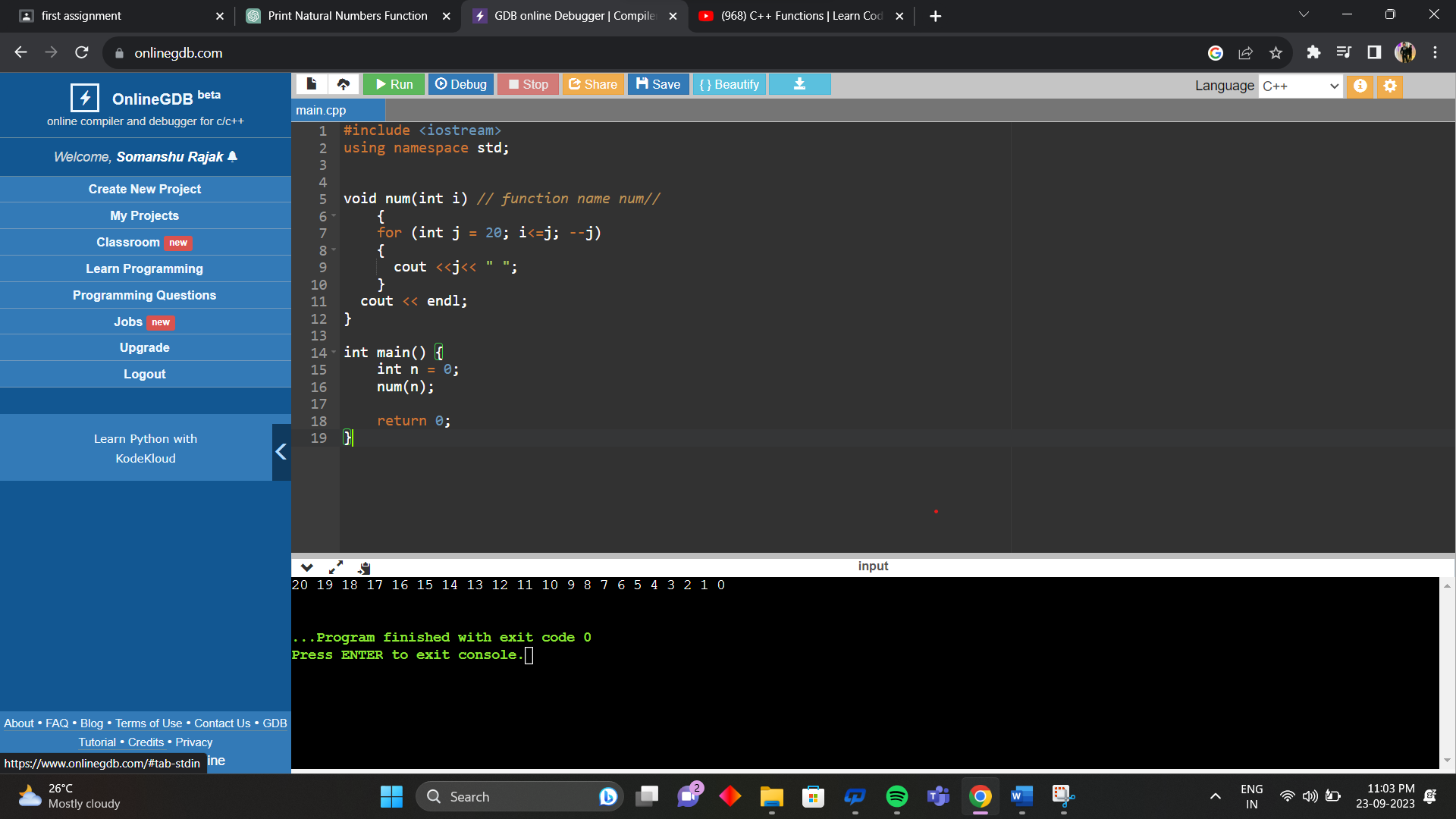
int main() {

int n = 0;

num(n);

return 0;

}



3. WAP for printing all even numbers from 1 to 20.

#include <iostream>

using namespace std;

void num(int i) // function name num//

{ {

cout<<"Even numbers are :";

}

for (int j = 0; i>=j; j+=2)

{

cout <<j<< " ";

}

cout << endl;

}

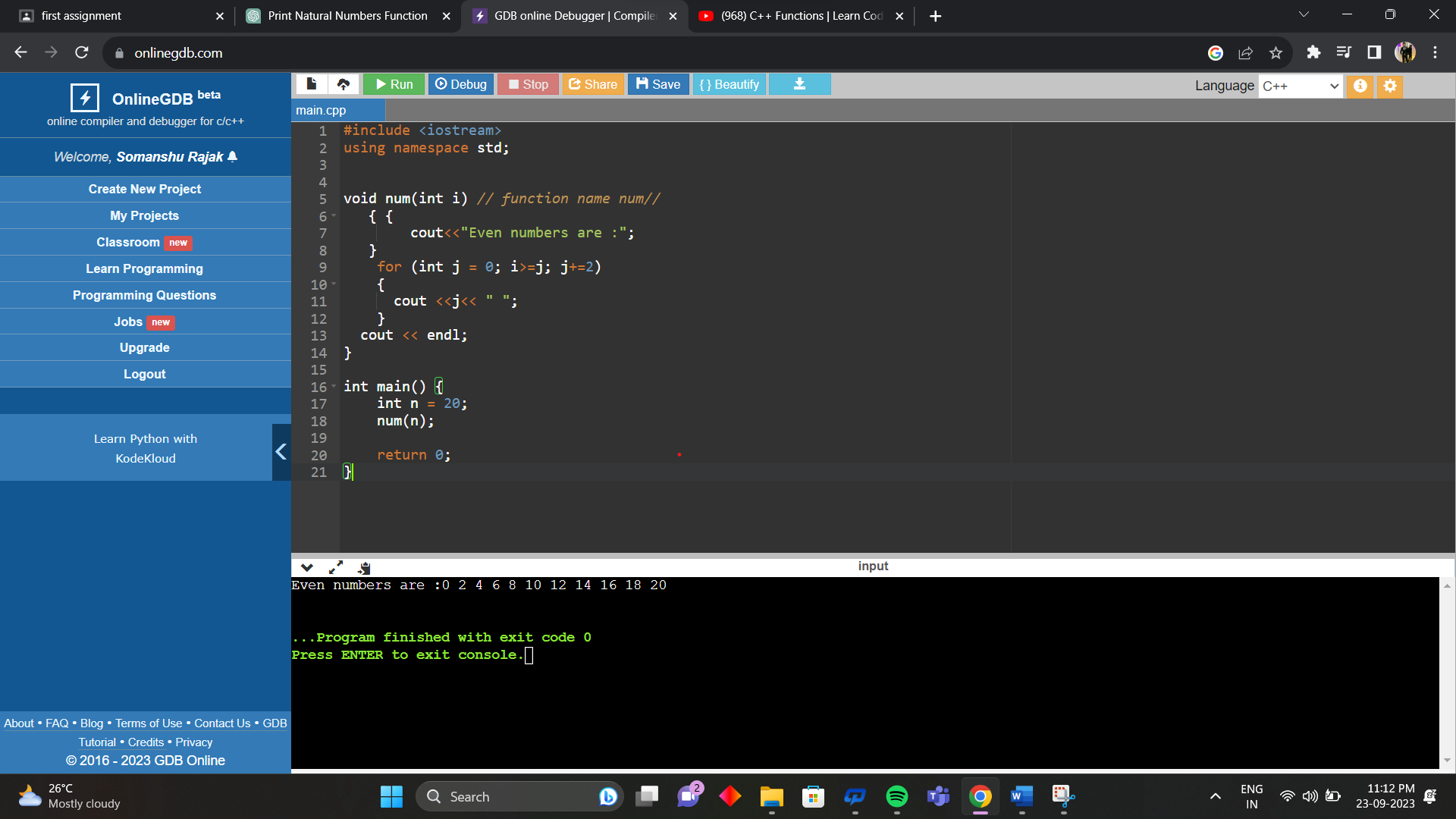
int main() {

int n = 20;

num(n);

return 0;

}



4. WAP for printing all odd numbers from 1 to 20.

#include <iostream>

using namespace std;

void num(int i) // function name num//

{ {

cout<<"Odd numbers are :";

}

for (int j = 1; i>=j; j+=2)

{

cout <<j<< " ";

}

cout << endl;

}

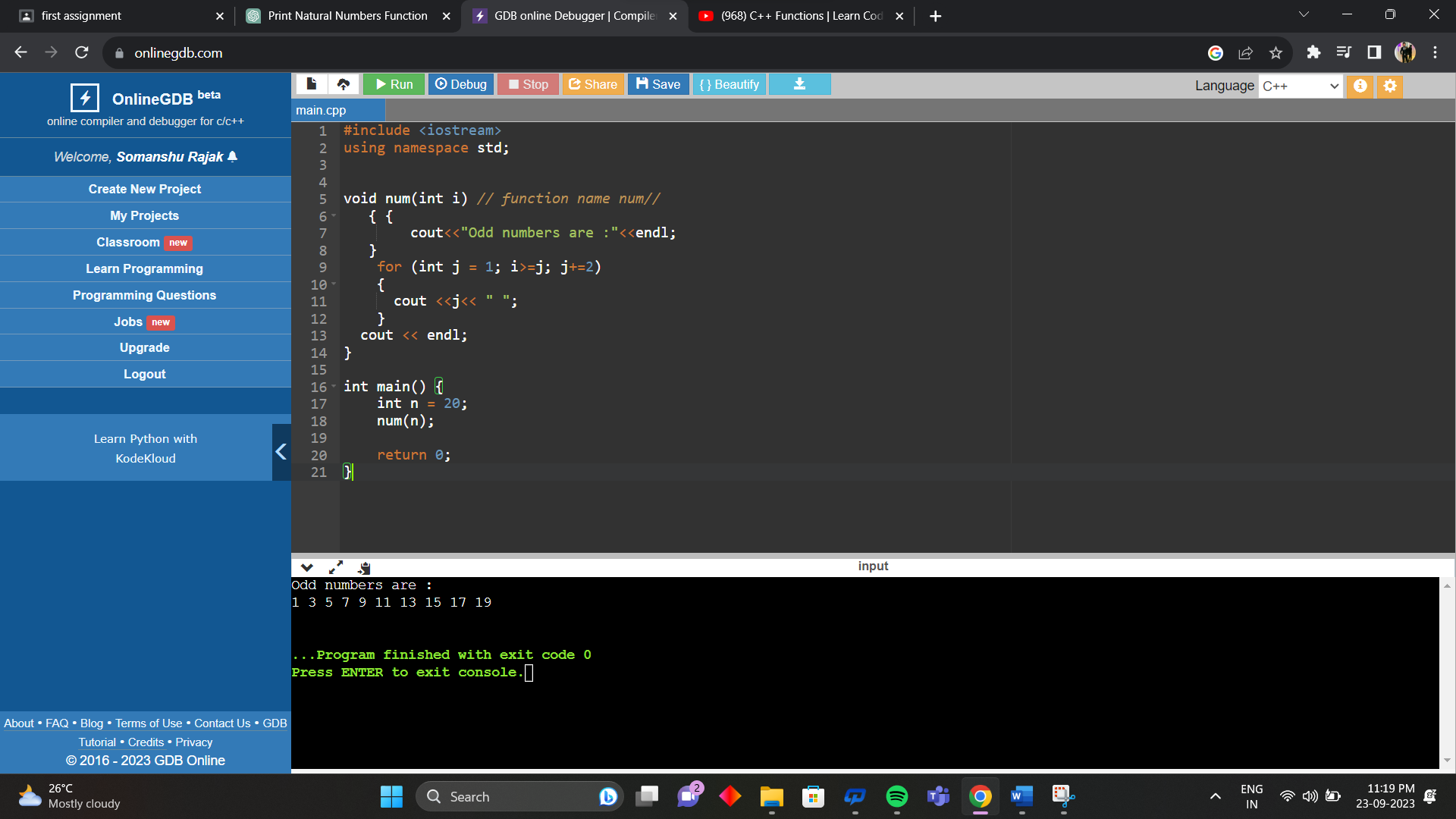
int main() {

int n = 20;

num(n);

return 0;

}



5. WAP for adding all numbers from 1 to 20.

#include <iostream>

using namespace std;

int num(int a, int b)

{

int sum = 0;

for (int i = a; i <= b; ++i) {

sum += i;

}

return sum;

}

int main()

{

int a=0;

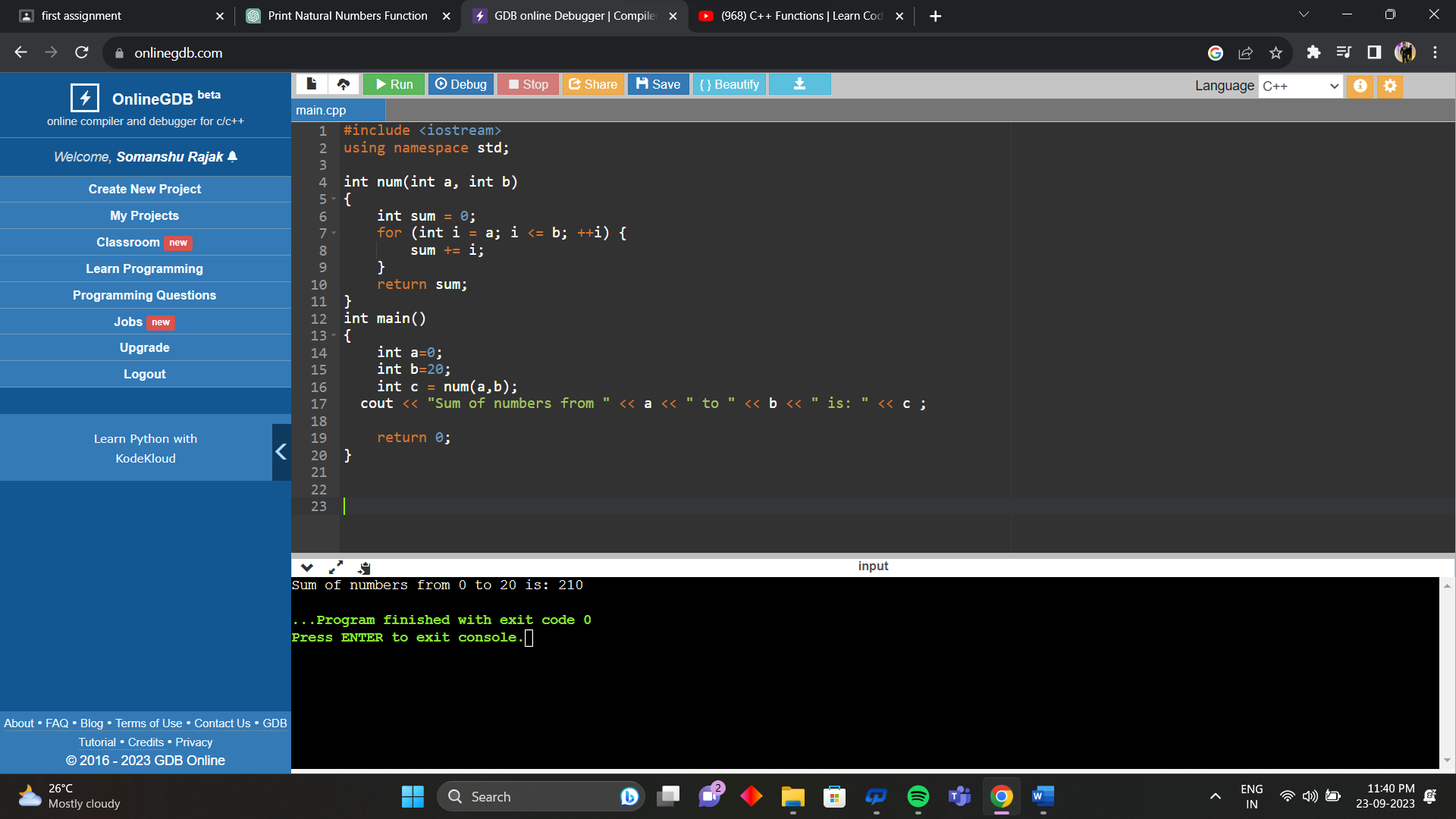
int b=20;

int c = num(a,b);

cout << "Sum of numbers from " << a << " to " << b << " is: " << c ;

return 0;

}



6. WAP for finding sum of all even numbers till 20.

#include <iostream>

using namespace std;

int num(int a)

{

int sum = 0;

for (int i = 2; i <= a; i += 2)

{

sum += i;

}

return sum;

}

int main()

{

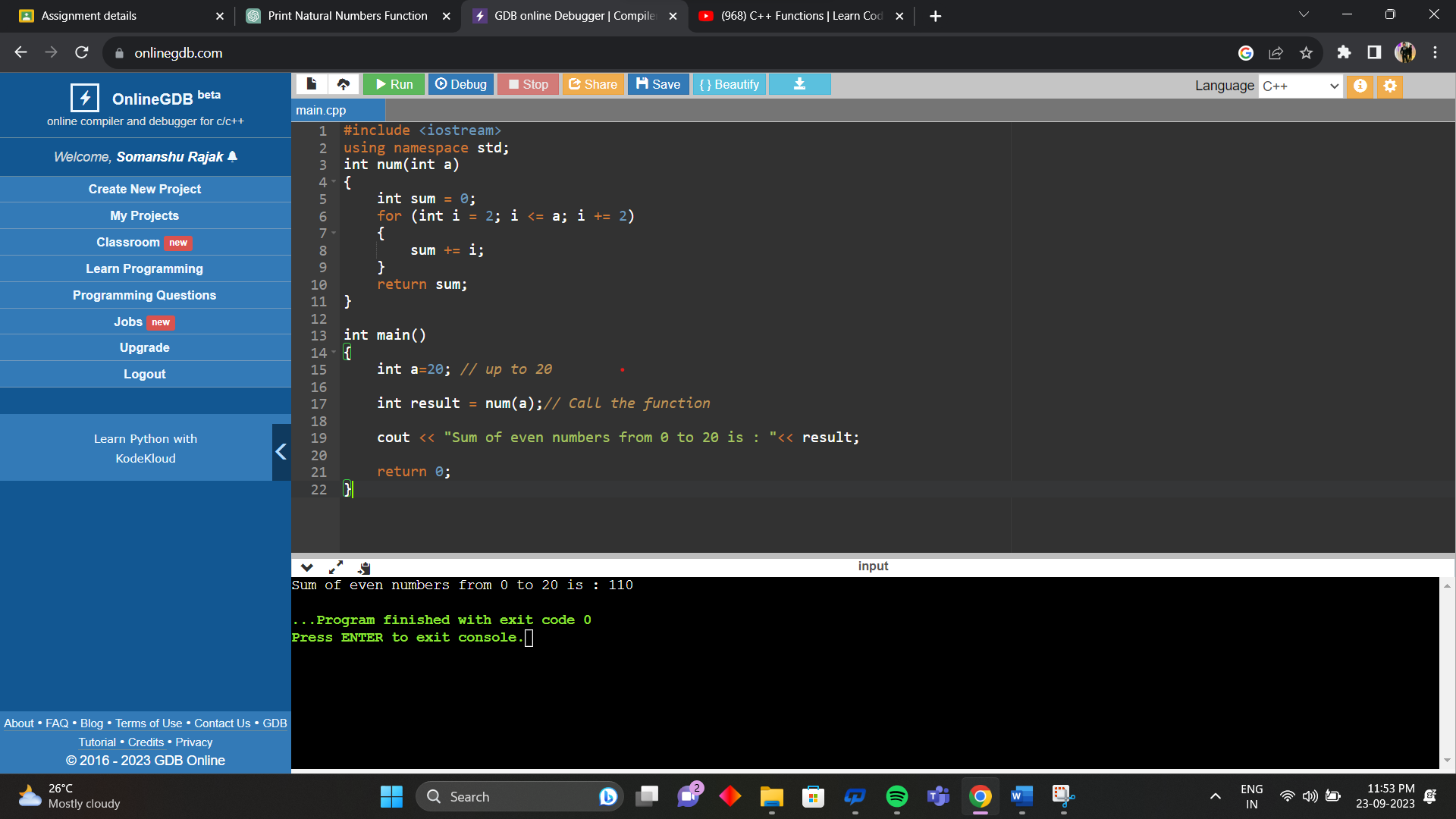
int a=20; // up to 20

int result = num(a);// Call the function

cout << "Sum of even numbers from 0 to 20 is : "<< result;

return 0;

}



7. WAP for finding sum of all odd numbers till 20.

#include <iostream>

using namespace std;

int num(int a)

{

int sum = 0;

for (int i = 1; i <= a; i += 2)

{

sum += i;

}

return sum;

}

int main()

{

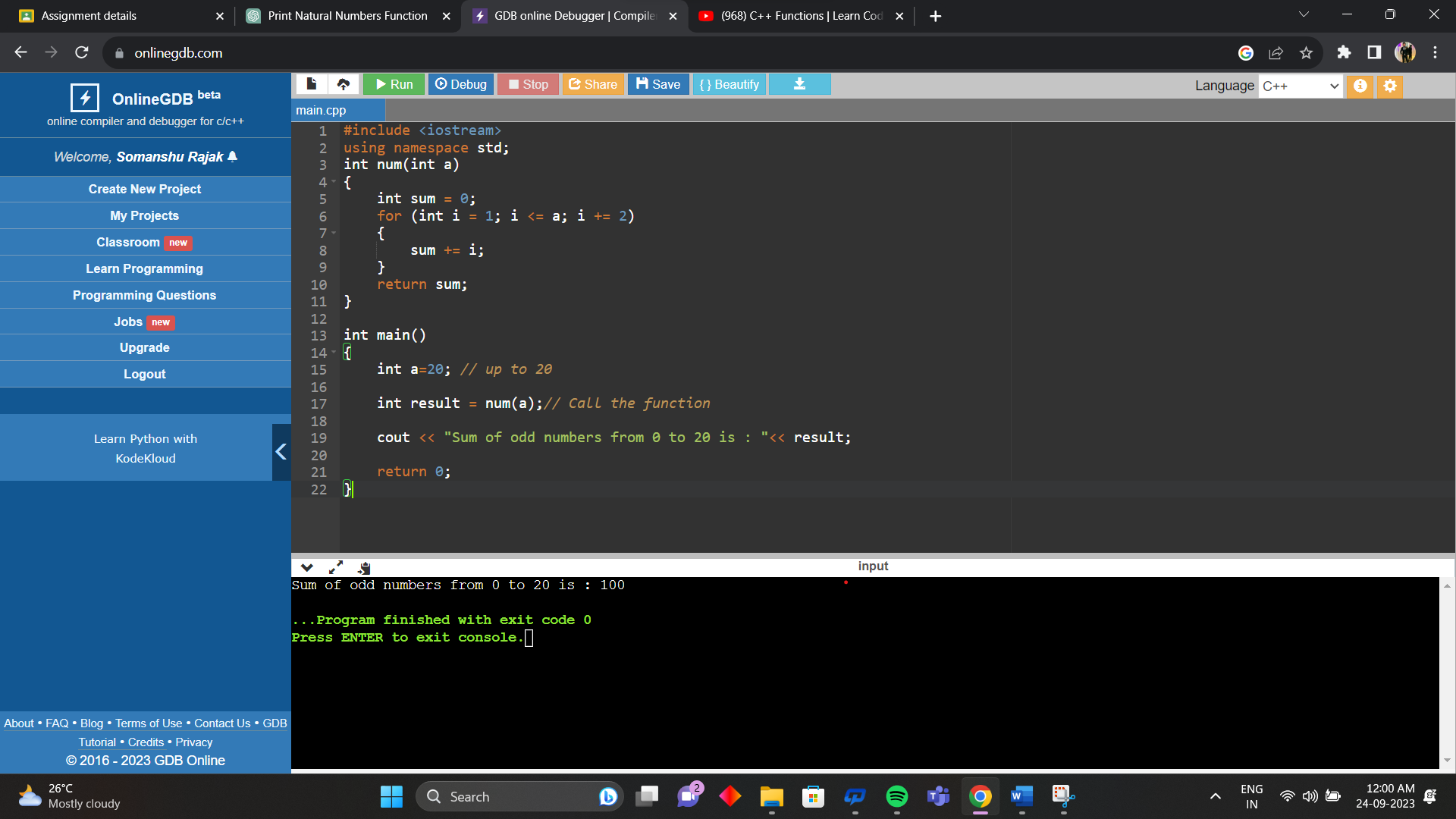
int a=20; // up to 20

int result = num(a);// Call the function

cout << "Sum of odd numbers from 0 to 20 is : "<< result;

return 0;

}



8. WAP for printing multiplication table of a number. For eg. Display should be “ 2 X 1 = 2”

#include <iostream>

using namespace std;

void num(int a)

{

for (int i = 1; i <= 10; ++i)

{

int b;

b= a \* i;

cout << a << " x " << i << " = " << b<<endl;

}

}

int main() {

int a;

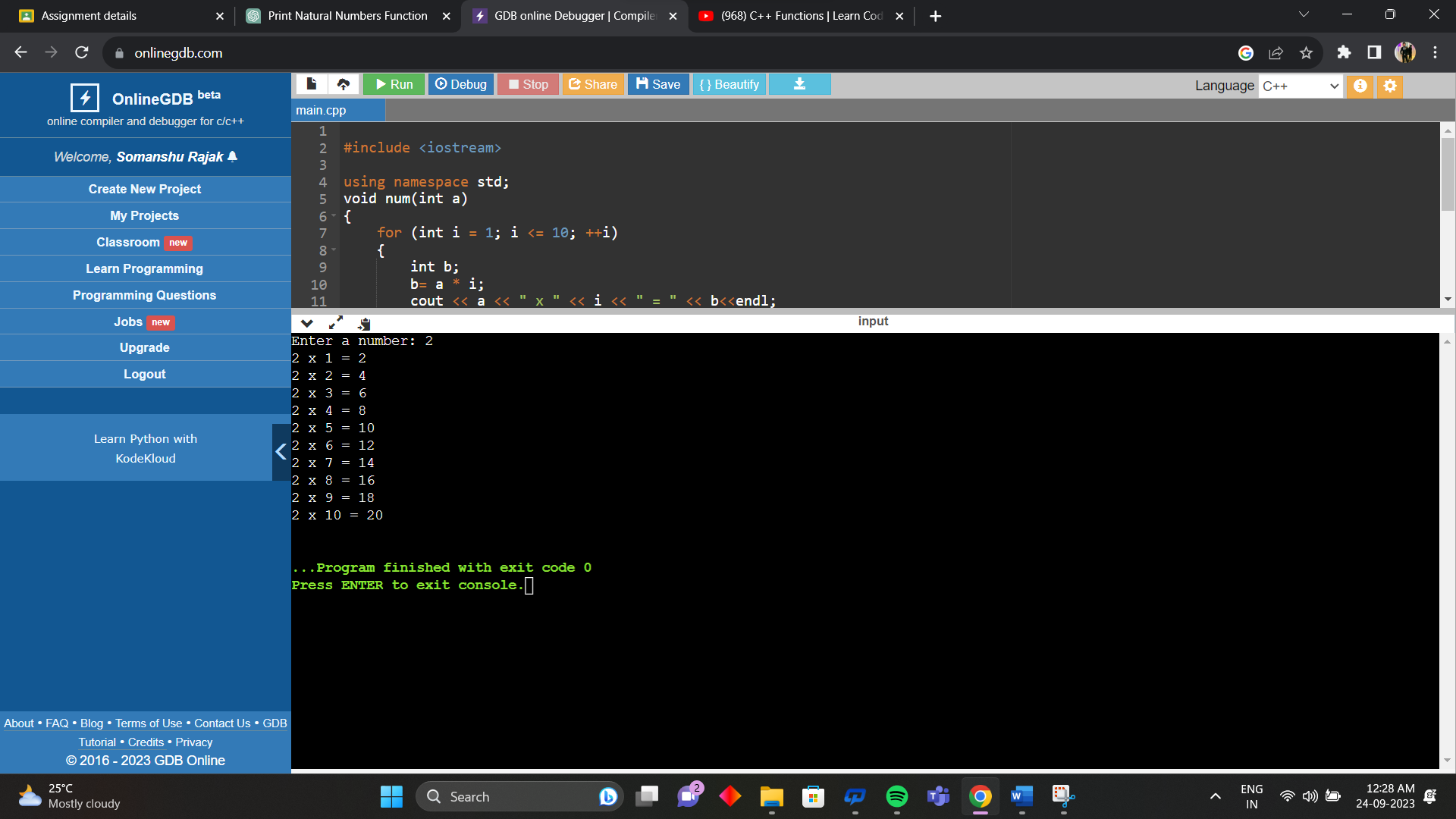
cout << "Enter a number: ";

cin >> a;

num(a);

return 0;

}



9. WAP to calculate factorial of a number.

#include <iostream>

using namespace std;

long factorial(int n)

{

if (n == 0 || n == 1)

{

return 1;

} else

{

return n \* factorial(n - 1);

}

}

int main()

{

int number;

cout << "Enter a number: ";

cin >> number;

if (number < 0) {

cout << "Factorial is always positive number ." << endl;

} else

{

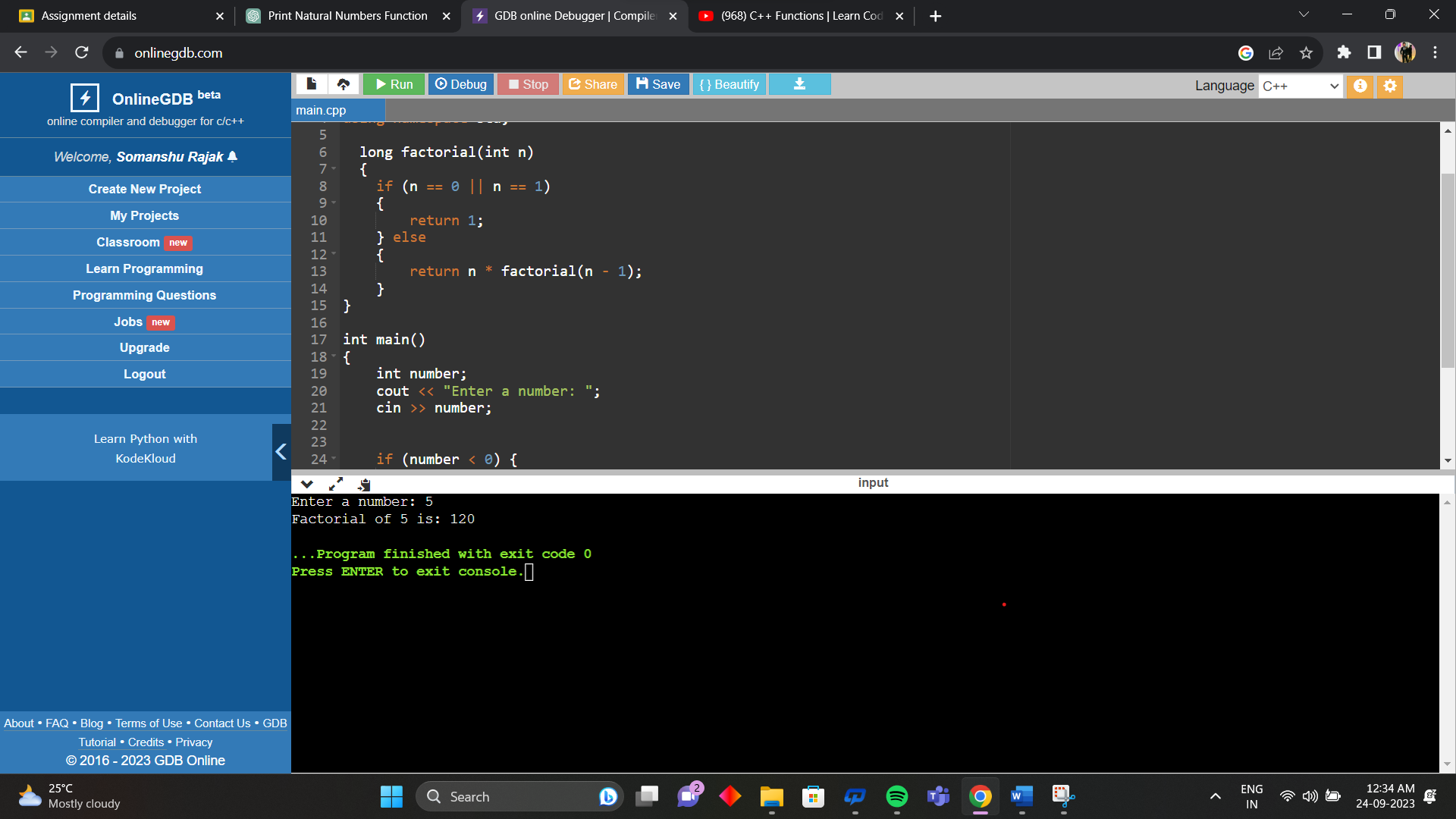
long result = factorial (number);

cout << "Factorial of " << number << " is: " << result;

}

return 0;

}



10. WAP to check whether a number is prime or not.

#include <iostream>

#include <math.h>

using namespace std;

bool isPrime(int number) {

if (number <= 1) {

return false;

}

if (number == 2) {

return true;

}

for (int i = 2; i <= sqrt(number); ++i)

{

if (number % i == 0) {

return false;

}

}

return true;

}

int main() {

int number;

cout << "Enter a number: ";

cin >> number;

if (isPrime(number)) {

cout << number << " is a prime number." << endl;

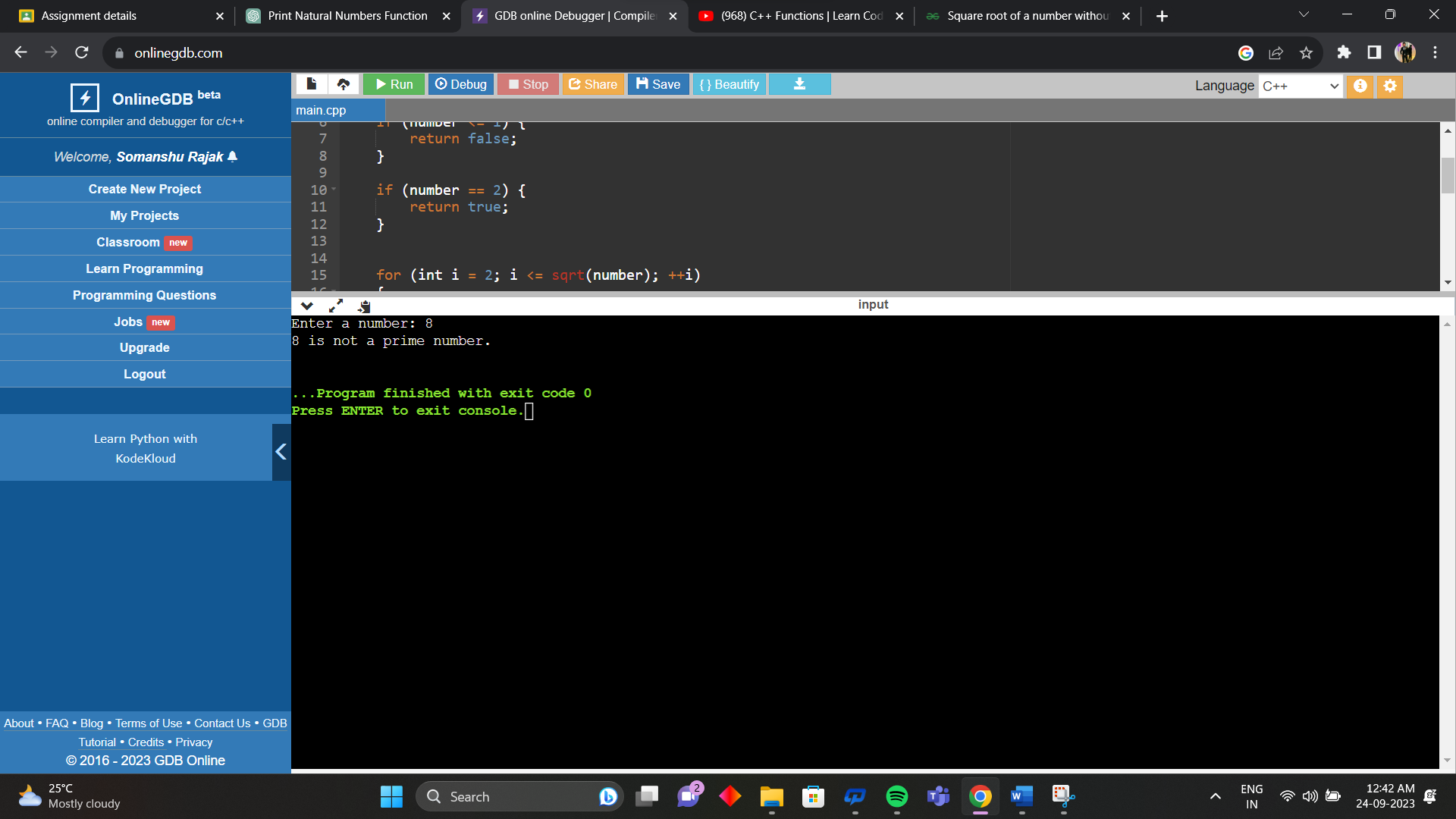
} else {

cout << number << " is not a prime number." << endl;

}

return 0;

}



11. WAP to print all digits of a number and their sum.

#include <iostream>

using namespace std;

void num(int a)

{

int sum = 0;

cout << "Digits are " <<" "<< a;

while (a > 0) {

int digit = a % 10;

cout << digit << " ";

sum += digit;

a /= 10;

}

cout << endl << "Sum of the digits is: " << sum << endl;

}

int main() {

int a;

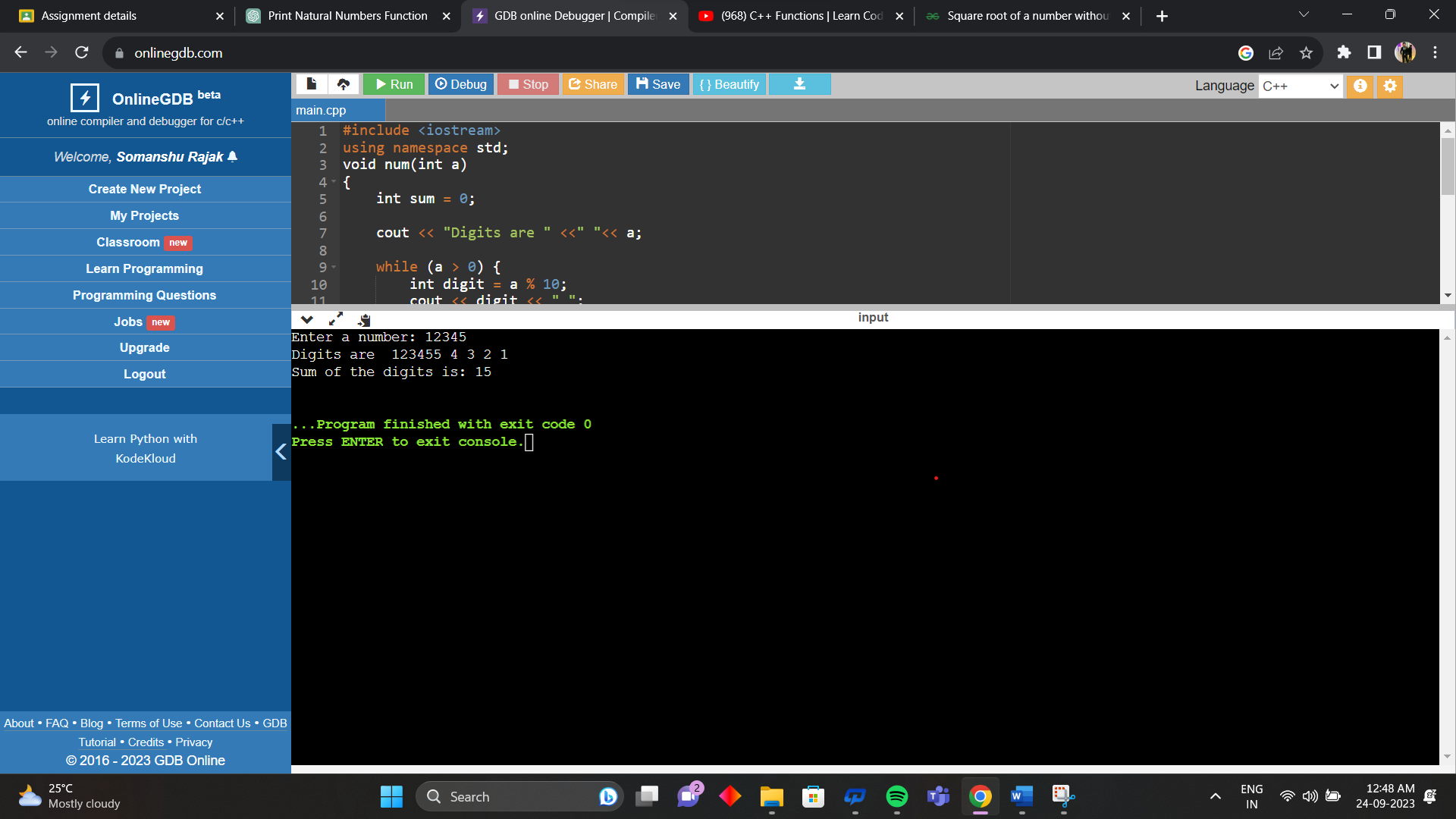
cout << "Enter a number: ";

cin >> a;

num(a);

return 0;

}



12. WAP to print reverse of a number.

#include <iostream>

using namespace std;

int reverseNumber(int number)

{

int reversed = 0;

while (number > 0)

{

int digit = number % 10;

reversed = reversed \* 10 + digit;

number /= 10;

}

return reversed;

}

int main() {

int number;

cout << "Enter a number: ";

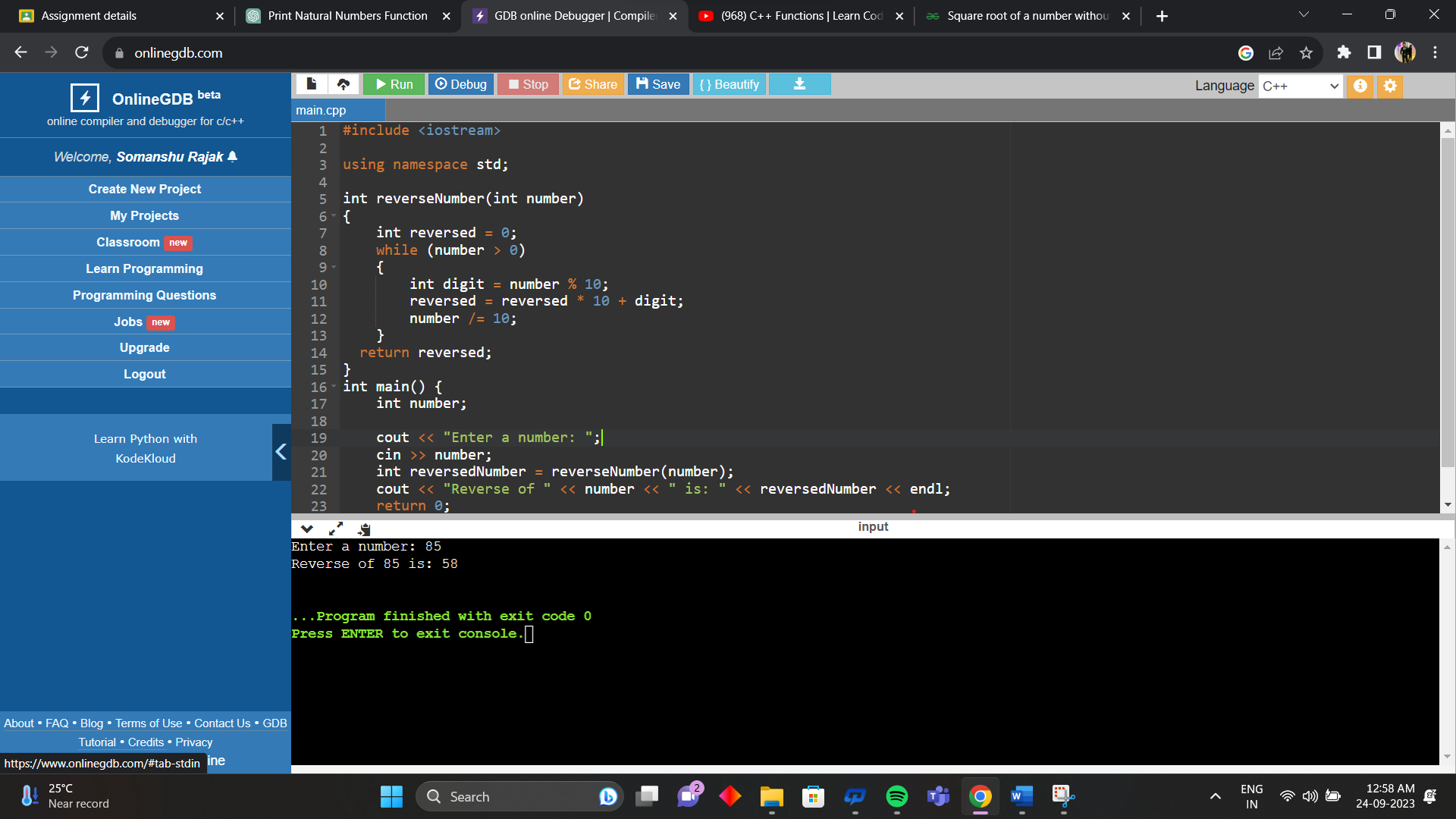
cin >> number;

int reversedNumber = reverseNumber(number);

cout << "Reverse of " << number << " is: " << reversedNumber << endl;

return 0;

}



13. WAP to check whether the number is Armstrong or not.

#include <iostream>

#include <cmath>

using namespace std;

int countDigits(int number) {

int count = 0;

while (number > 0) {

count++;

number /= 10;

}

return count;

}

bool isArmstrong(int number) {

int originalNumber = number;

int numDigits = countDigits(number);

int sum = 0;

while (number > 0) {

int digit = number % 10;

sum += pow(digit, numDigits);

number /= 10;

}

return (sum == originalNumber);

}

int main() {

int number;

cout << "Enter a number: ";

cin >> number;

if (isArmstrong(number))

{

cout << number << " is an Armstrong number." << endl;

}

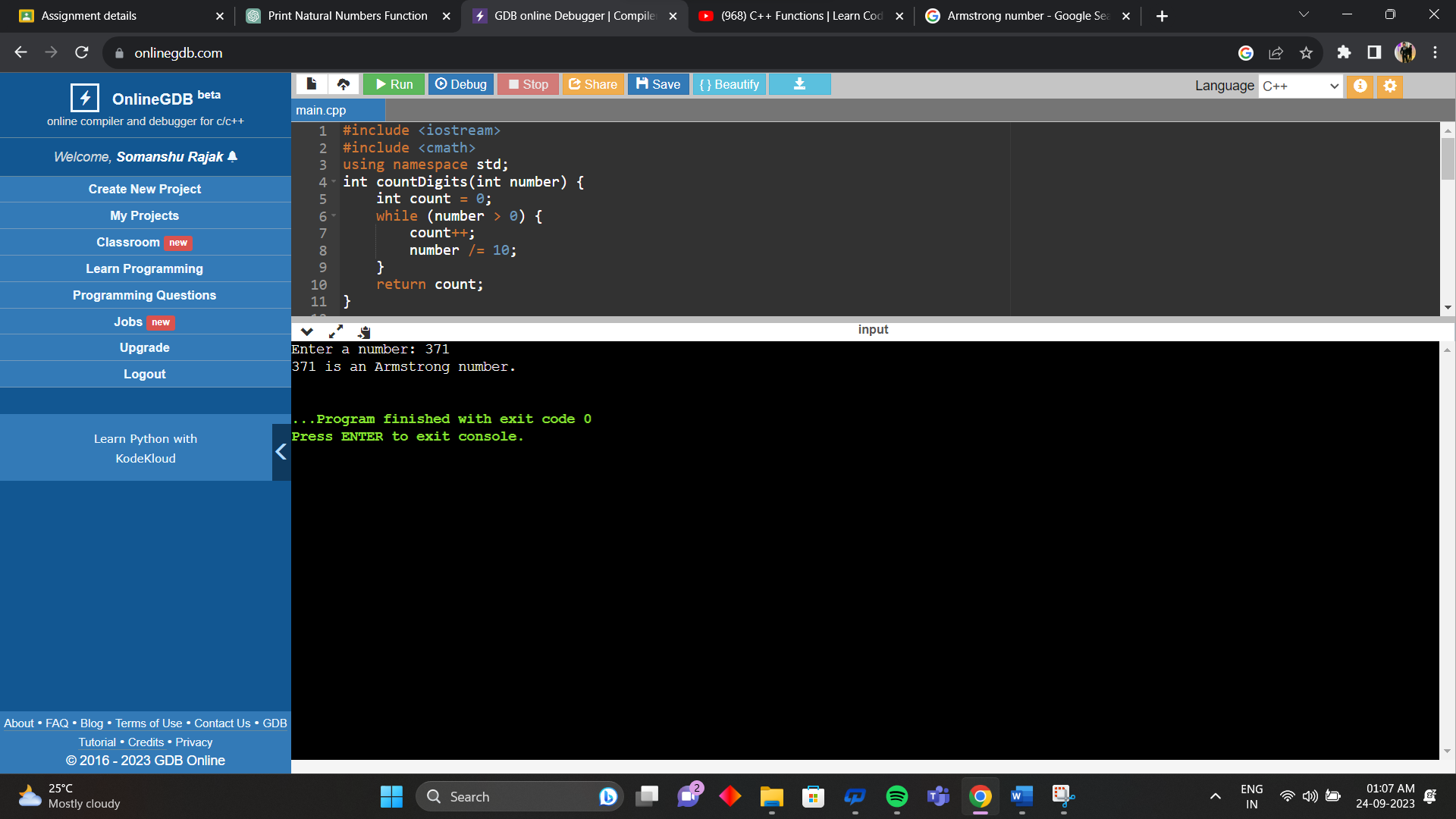
else {

cout << number << " is not an Armstrong number." << endl;

}

return 0;

}



14. WAP to print the Fibonacci series in a given range.

#include <iostream>

using namespace std;

void printFibonacciSeries(int range) {

int first = 0, second = 1, next;

cout << "Fibonacci series up to " << range << " terms: ";

while (first <= range) {

cout << first << " ";

next = first + second;

first = second;

second = next;

}

cout << endl;

}

int main() {

int range;

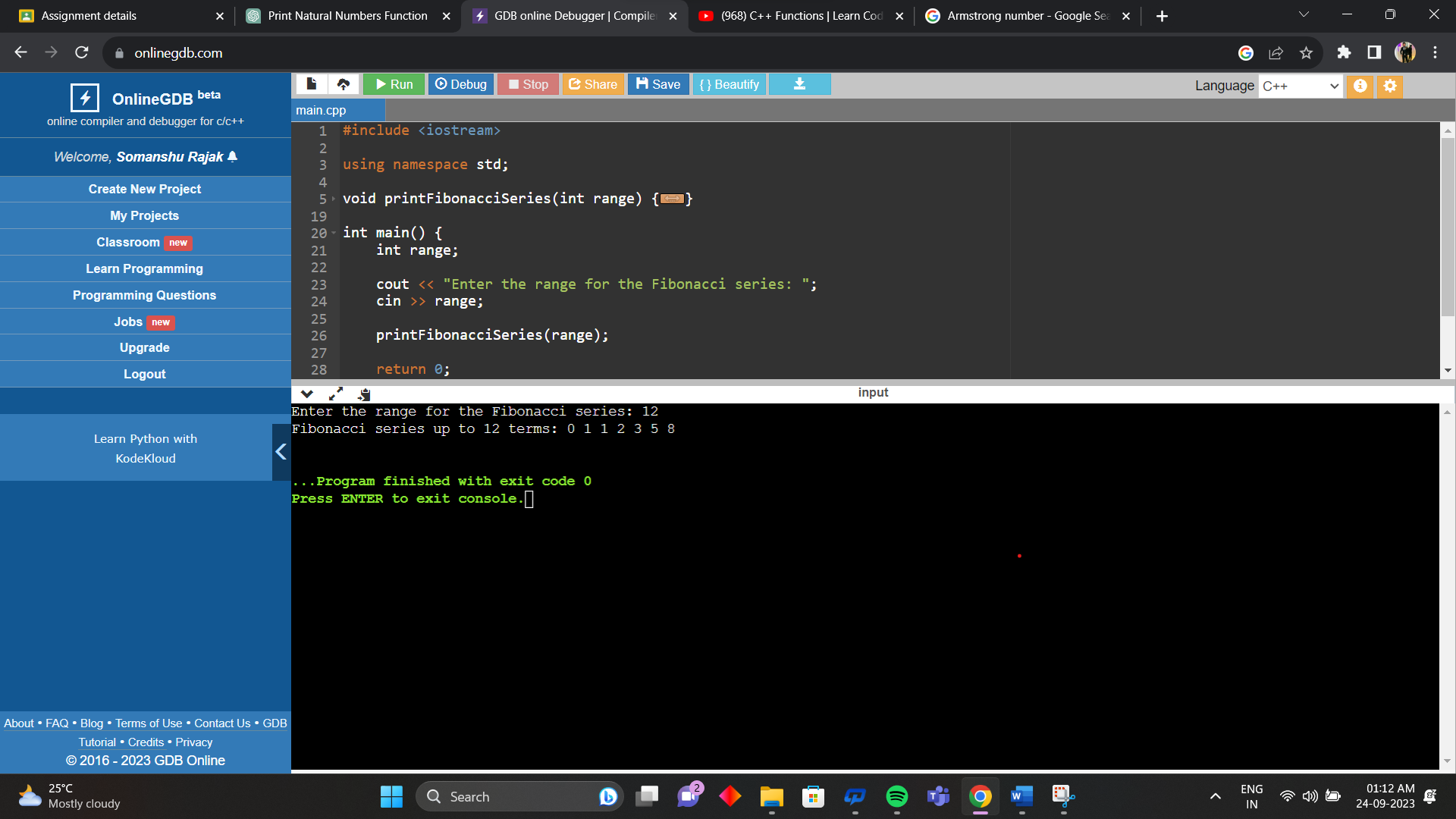
cout << "Enter the range for the Fibonacci series: ";

cin >> range;

printFibonacciSeries(range);

return 0;

}



15. WAP to check whether the number entered is palindrome or not.

#include <iostream>

using namespace std;

bool isPalindrome(int number) {

int originalNumber = number;

int reversedNumber = 0;

while (number > 0) {

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number /= 10;

}

return (reversedNumber == originalNumber);

}

int main() {

int number;

cout << "Enter a number: ";

cin >> number;

if (isPalindrome(number)) {

cout << number << " is a palindrome number." << endl;

} else {

cout << number << " is not a palindrome number." << endl;

}

return 0;

}

