APP WEEK 3 ASSIGNMENT

**1) Simple mini calculator program in C++ that uses subroutines for basic arithmetic operations**

#include <iostream>

// Function to add two numbers

int add(int a,int b){

return a+b;

}

// Function to subtract two numbers

int subtract(int a,int b){

return a-b;

}

// Function to multiply two numbers

int multiply(int a,int b){

return a\*b;

}

// Function to divide two numbers

int divide(int a,int b) {

if (b==0) {

std::cout<<"Error: Cannot divide by zero!\n";

return 0;

}

return a/b;

}

int main(){

int num1,num2;

char op;

std::cout<<"Enter two numbers:";

std::cin>>num1>>num2;

std::cout<<"Enter an operator(+, -, \*, /):";

std::cin>>op;

int result;

switch(op){

case '+':

result=add(num1, num2);

break;

case '-':

result=subtract(num1, num2);

break;

case '\*':

result=multiply(num1, num2);

break;

case '/':

result=divide(num1, num2);

break;

default:

std::cout<<"Invalid operator!\n";

return 1;

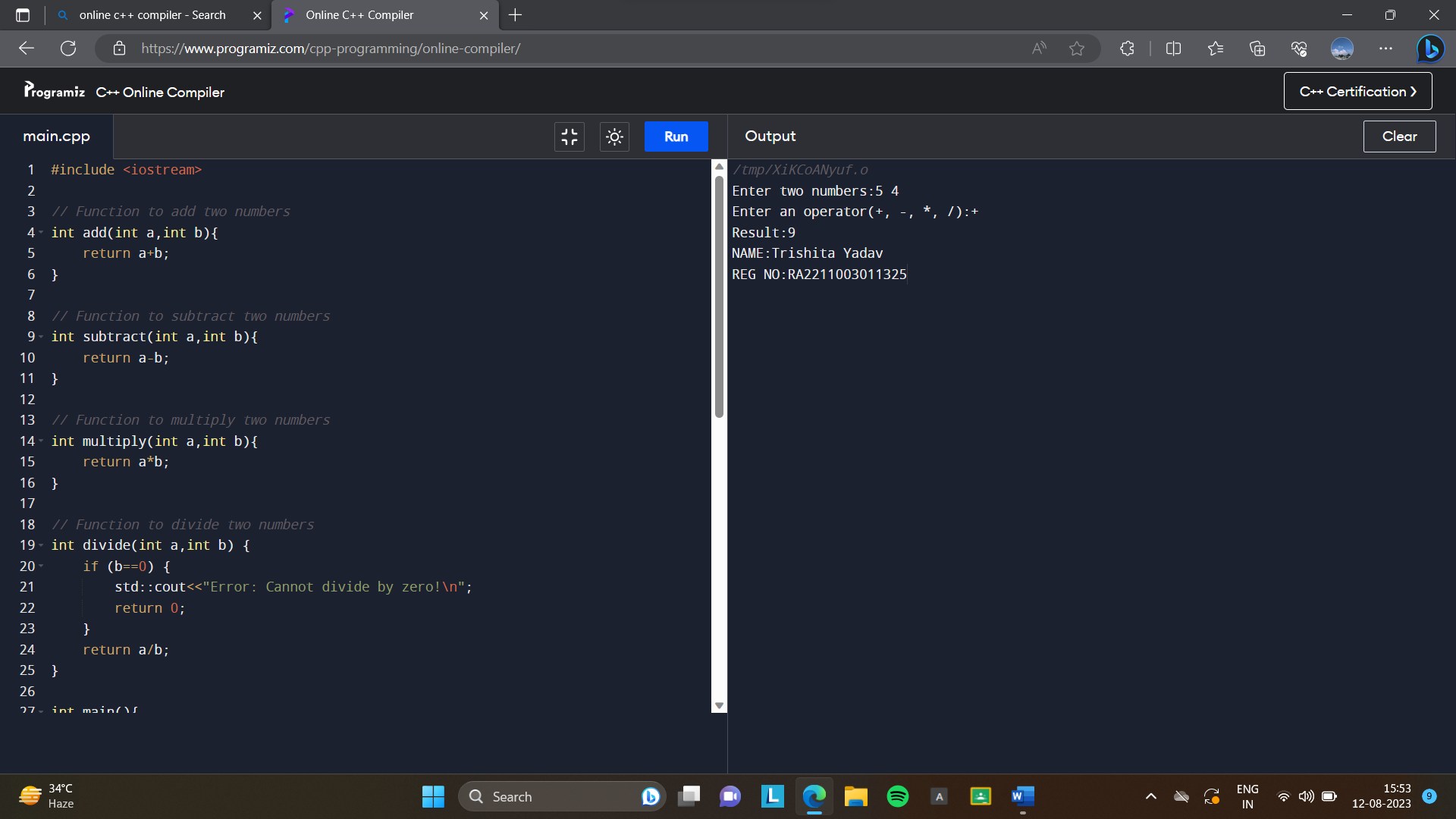
}

std::cout<<"Result:"<<result<<std::endl;

std::cout<<"NAME:Trishita Yadav\nREG NO:RA2211003011325";

return 0;

}



**2) Write a complete students database with subroutines involves storing and managing student information using appropriate data structures and providing various functionalities to interact with the database and implement in C++ with subroutines:**

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Variable to keep track of no. of students

int i=0;

// Structure to store the student

struct sinfo{

char fname[50];

char lname[50];

int roll;

float cgpa;

}st[55];

// Function to add the student

void add\_student()

{

printf("Add the Students Details\n");

printf("-------------------------\n");

printf("Enter the first name of student:\n");

scanf("%s",st[i].fname);

printf("Enter the last name of student:\n");

scanf("%s",st[i].lname);

printf("Enter the Roll Number:\n");

scanf("%d",&st[i].roll);

printf("Enter the CGPA you obtained:\n");

scanf("%f",&st[i].cgpa);

i = i + 1;

}

// Function to find student by roll number

void find\_rl()

{

int x;

printf("Enter the Roll Number of the student:\n");

scanf("%d",&x);

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

if(x==st[i].roll){

printf("The Students Details are:\n");

printf("The First name is %s\n",st[i].fname);

printf("The Last name is %s\n",st[i].lname);

printf("The CGPA is %f\n",st[i].cgpa);

}

break;

}

}

// Function to find the student by the first name

void find\_fn()

{

char a[50];

printf("Enter the First Name of the student:\n");

scanf("%s", a);

int c = 0;

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

if (!strcmp(st[j].fname, a)){

printf("The Students Details are:\n");

printf("The First name is %s\n",st[i].fname);

printf("The Last name is %s\n",st[i].lname);

printf("The Roll Number is %d\n ",st[i].roll);

printf("The CGPA is %f\n",st[i].cgpa);

c = 1;

}

else

printf("The First Name not Found.\n");

}

}

// Function to print the total number of students

void tot\_s()

{

printf("The total number of Student is %d\n",i);

printf("\n you can have a max of 50 students\n");

printf("you can have %d more students\n",50 - i);

}

// Function to delete a student by the roll number

void del\_s()

{

int a;

printf("Enter the Roll Number which you want to delete:\n");

scanf("%d",&a);

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

if(a==st[j].roll){

for(int k=j;k<49;k++)

st[k]=st[k + 1];

i--;

}

}

printf("The Roll Number is removed Successfully.\n");

}

// Function to update a students data

void up\_s()

{

printf("Enter the roll number to update the entry:");

long int x;

scanf("%ld",&x);

for(int j=0;j<i;j++){

if(st[j].roll==x){

printf("1.First name\n"

"2.Last name\n"

"3.Roll no.\n"

"4.CGPA\n");

int z;

scanf("%d",&z);

switch(z){

case 1:

printf("Enter the new first name:\n");

scanf("%s",st[j].fname);

break;

case 2:

printf("Enter the new last name:\n");

scanf("%s",st[j].lname);

break;

case 3:

printf("Enter the new roll number: \n");

scanf("%d", &st[j].roll);

break;

case 4:

printf("Enter the new CGPA:\n");

scanf("%f", &st[j].cgpa);

break;

}

printf("UPDATED SUCCESSFULLY.\n");

}

}

}

int main()

{

printf("NAME:Trishita Yadav\nREG NO:RA2211003011325");

int choice, count;

while(i = 1){

printf("\nThe Task that you want to perform\n");

printf("1. Add the Student Details\n");

printf("2. Find the Student Details by Roll Number\n");

printf("3. Find the Student Details by First Name\n");

printf("4. Find the Total number of Students\n");

printf("5. Delete the Students Details by Roll Number\n");

printf("6. Update the Students Details by Roll Number\n");

printf("7. To Exit\n");

printf("Enter your choice to find the task\n");

scanf("%d", &choice);

switch(choice){

case 1:

add\_student();

break;

case 2:

find\_rl();

break;

case 3:

find\_fn();

break;

case 4:

tot\_s();

break;

case 5:

del\_s();

break;

case 6:

up\_s();

break;

case 7:

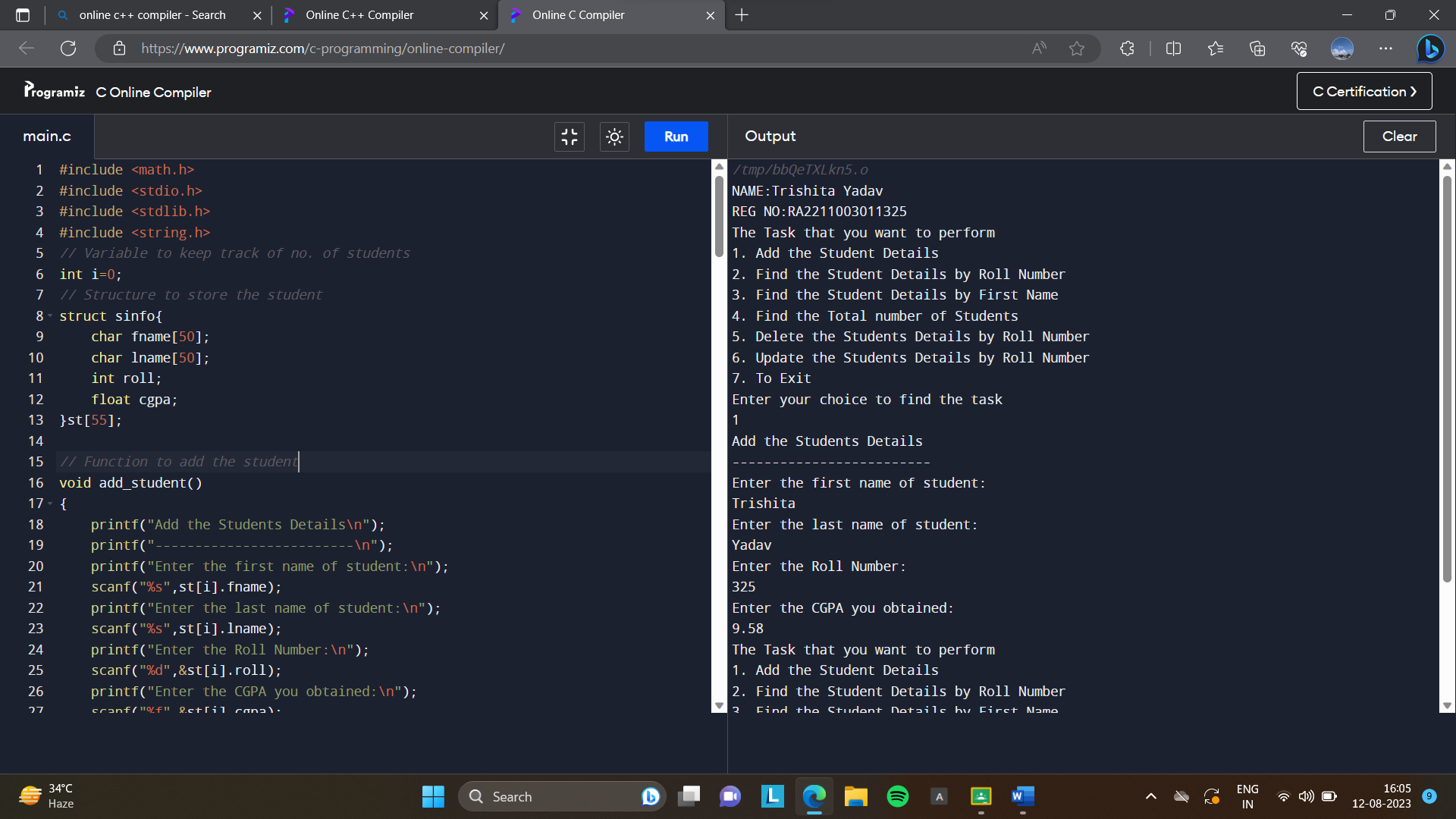
exit(0);

break;

}

}

}



**3) Design a subroutine program to calculate the area and perimeter of different geometric**

**shapes (circle, rectangle, triangle, etc.).**

#include <iostream>

#include <cmath>

// Function to calculate the area and perimeter of a circle

void circleProperties(double radius,double& area,double& perimeter){

const double pi=3.14159;

area=pi\*radius\*radius;

perimeter=2\*pi\*radius;

}

// Function to calculate the area and perimeter of a rectangle

void rectangleProperties(double length,double width,double& area,double& perimeter){

area=length\*width;

perimeter=2\*(length+width);

}

// Function to calculate the area and perimeter of a triangle

void triangleProperties(double side1,double side2,double side3,double& area,double& perimeter) {

double s=(side1+side2+side3)/2;

area=std::sqrt(s\*(s-side1)\*(s-side2)\*(s-side3));

perimeter=side1+side2+side3;

}

int main() {

std::cout<<"NAME:Trishita Yadav\nREG NO:RA2211003011325";

int choice;

double area,perimeter;

do{

std::cout<<"\nSelect a shape to calculate properties:\n";

std::cout<<"1.Circle\n";

std::cout<<"2.Rectangle\n";

std::cout<<"3.Triangle\n";

std::cout<<"4.Exit\n";

std::cout<<"Enter your choice:";

std::cin>>choice;

switch(choice){

case 1:{

double radius;

std::cout<<"Enter the radius of the circle:";

std::cin>>radius;

circleProperties(radius,area,perimeter);

std::cout<<"Area:"<<area<<"\n";

std::cout<<"Perimeter:"<<perimeter<<"\n";

break;

}

case 2:{

double length,width;

std::cout<<"Enter the length of the rectangle:";

std::cin>>length;

std::cout<<"Enter the width of the rectangle:";

std::cin>>width;

rectangleProperties(length,width,area,perimeter);

std::cout <<"Area:"<<area<<"\n";

std::cout <<"Perimeter:"<<perimeter<<"\n";

break;

}

case 3:{

double side1,side2,side3;

std::cout<<"Enter the length of side 1 of the triangle:";

std::cin>>side1;

std::cout<<"Enter the length of side 2 of the triangle:";

std::cin>>side2;

std::cout<<"Enter the length of side 3 of the triangle:";

std::cin>>side3;

triangleProperties(side1,side2,side3,area,perimeter);

std::cout<<"Area:"<<area<<"\n";

std::cout<<"Perimeter:"<<perimeter<<"\n";

break;

}

case 4:

std::cout<<"Exiting the program.\n";

break;

default:

std::cout<<"Invalid choice. Try again.\n";

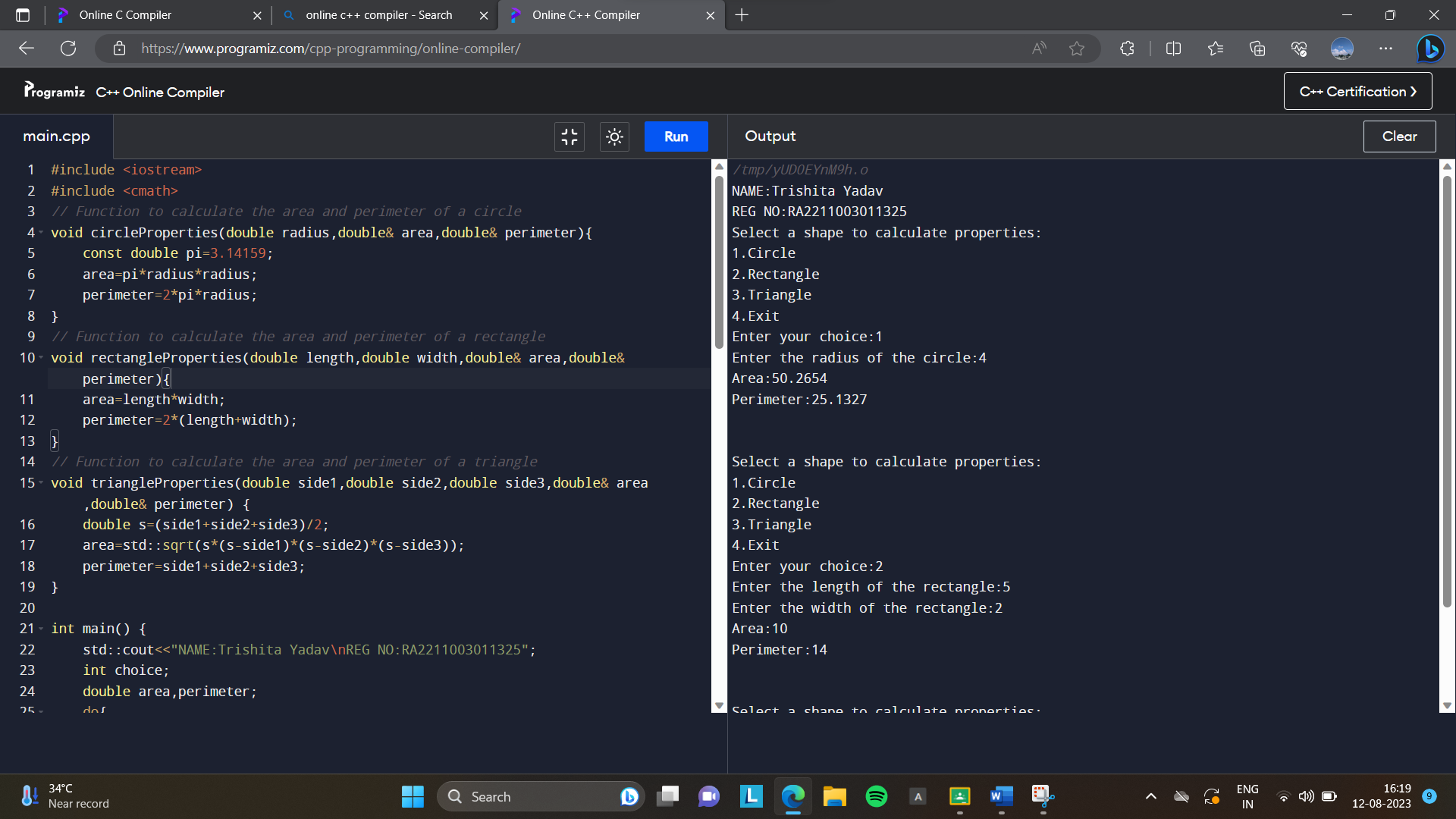
}

std::cout<<std::endl;

} while(choice!=4);

return 0;

}



**4) Implement a subroutine program to check if a given string is a palindrome or not.**

#include <iostream>

#include <string>

#include <cctype>

// Function to check if a string is a palindrome

bool isPalindrome(const std::string& str){

int left=0;

int right=str.length()-1;

while(left<right){

// Ignore non-alphanumeric characters from both ends

while(left<right && !std::isalnum(str[left])){

left++;

}

while(left<right && !std::isalnum(str[right])){

right--;

}

if(std::tolower(str[left]) != std::tolower(str[right])){

return false; // Characters do not match, not a palindrome

}

left++;

right--;

}

return true; // All characters match, it's a palindrome

}

int main(){

std::cout<<"NAME:Trishita Yadav\nREG NO:RA2211003011325";

std::string input;

std::cout<<"\nEnter a string:";

std::getline(std::cin, input);

if (isPalindrome(input)){

std::cout<<"The string is a palindrome.\n";

}

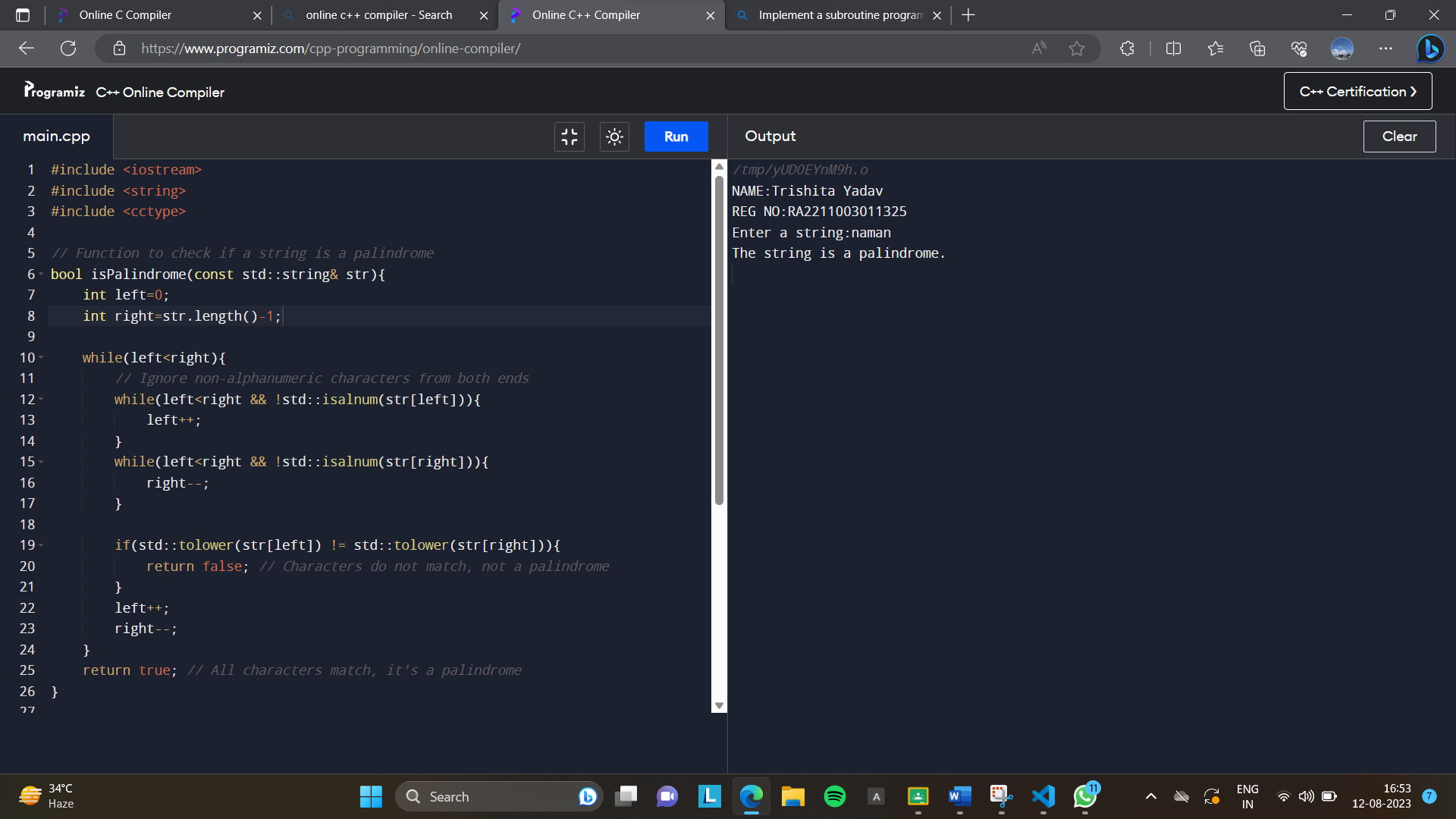
else{

std::cout<<"The string is not a palindrome.\n";

}

return 0;

}



**5) Implement a subroutine program to reverse an array of integers in-place**.

#include <iostream>

// Function to reverse an array of integers in-place

void reverseArray(int arr[],int size){

int left=0;

int right=size-1;

while(left<right){

// Swap elements at left and right indices

int temp=arr[left];

arr[left]=arr[right];

arr[right]=temp;

left++;

right--;

}

}

int main(){

std::cout<<"NAME:Trishita Yadav\nREG NO:RA2211003011325";

int size;

std::cout<<"\nEnter the size of the array:";

std::cin>>size;

int\* arr=new int[size];

std::cout<<"Enter "<<size<<" integers:";

for(int i=0;i<size;i++){

std::cin>>arr[i];

}

std::cout<<"Original array:";

for(int i=0;i<size;i++){

std::cout<<arr[i]<<" ";

}

std::cout<<std::endl;

reverseArray(arr,size);

std::cout<<"Reversed array:";

for(int i=0;i<size;i++){

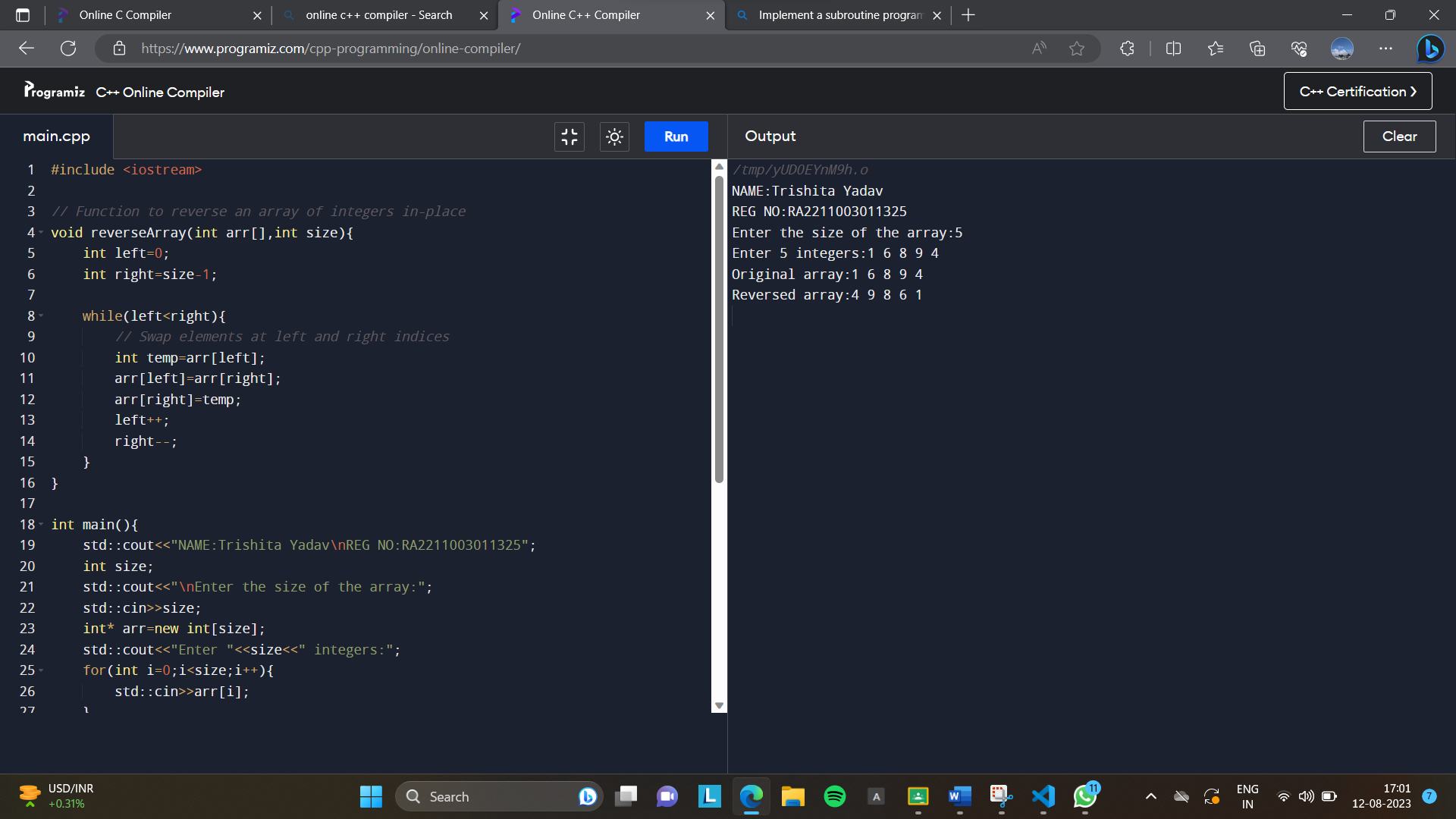
std::cout<<arr[i]<<" ";

}

std::cout<<std::endl;

delete[] arr;

return 0;

}

**6) Write a program that dynamically allocates memory for an array of integers based on user**

**input and then finds the sum of all elements in the array.**

#include <iostream>

int main(){

std::cout<<"NAME:Trishita Yadav\nREG NO:RA2211003011325";

int size;

std::cout<<"\nEnter the size of the array:";

std::cin>>size;

// Dynamically allocate memory for the array

int\* arr=new int[size];

std::cout<<"Enter "<<size<<" integers:";

for(int i=0;i<size;i++){

std::cin>>arr[i];

}

// Calculate the sum of elements in the array

int sum=0;

for(int i=0;i<size;i++){

sum+=arr[i];

}

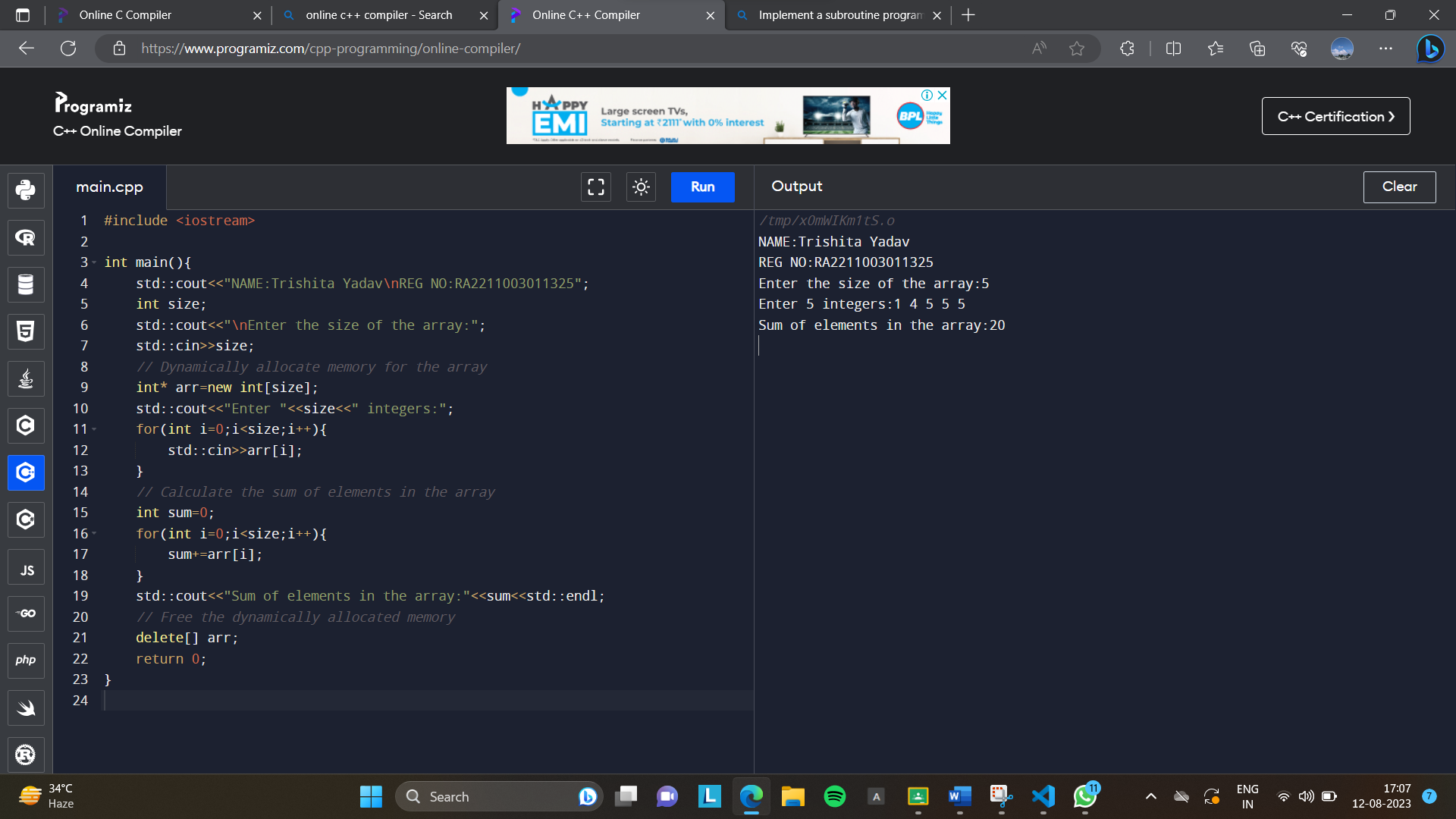
std::cout<<"Sum of elements in the array:"<<sum<<std::endl;

// Free the dynamically allocated memory

delete[] arr;

return 0;

}



**8) Create a program that uses dynamic memory allocation to implement a stack data structure**

**to push and pop elements.**

//Stack implementation in Java

class Stack{

  //store elements of stack

  private int arr[];

  //represent top of stack

  private int top;

  //total capacity of the stack

  private int capacity;

  //Creating a stack

  Stack(int size) {

    //initialize the array

    //initialize the stack variables

    arr = new int[size];

    capacity = size;

    top = -1;

  }

  //push elements to the top of stack

  public void push(int x){

    if(isFull()){

      System.out.println("Stack OverFlow");

      // terminates the program

      System.exit(1);

    }

    //insert element on top of stack

    System.out.println("Inserting " + x);

    arr[++top]=x;

  }

  //pop elements from top of stack

  public int pop(){

    //if stack is empty

    //no element to pop

    if(isEmpty()){

      System.out.println("STACK EMPTY");

      //terminates the program

      System.exit(1);

    }

    //pop element from top of stack

    return arr[top--];

  }

  //return size of the stack

  public int getSize(){

    return top + 1;

  }

  //check if the stack is empty

  public Boolean isEmpty(){

    return top==-1;

  }

  //check if the stack is full

  public Boolean isFull(){

    return top==capacity-1;

  }

  //display elements of stack

  public void printStack(){

    for(int i = 0; i <= top; i++){

      System.out.print(arr[i] +", ");

    }

  }

  public static void main(String[] args){

    Stack stack=new Stack(5);

    System.out.println("NAME:Trishita Yadav\nREG NO:RA2211003011325");

    stack.push(1);

    stack.push(2);

    stack.push(3);

    System.out.print("Stack: ");

    stack.printStack();

    // remove element from stack

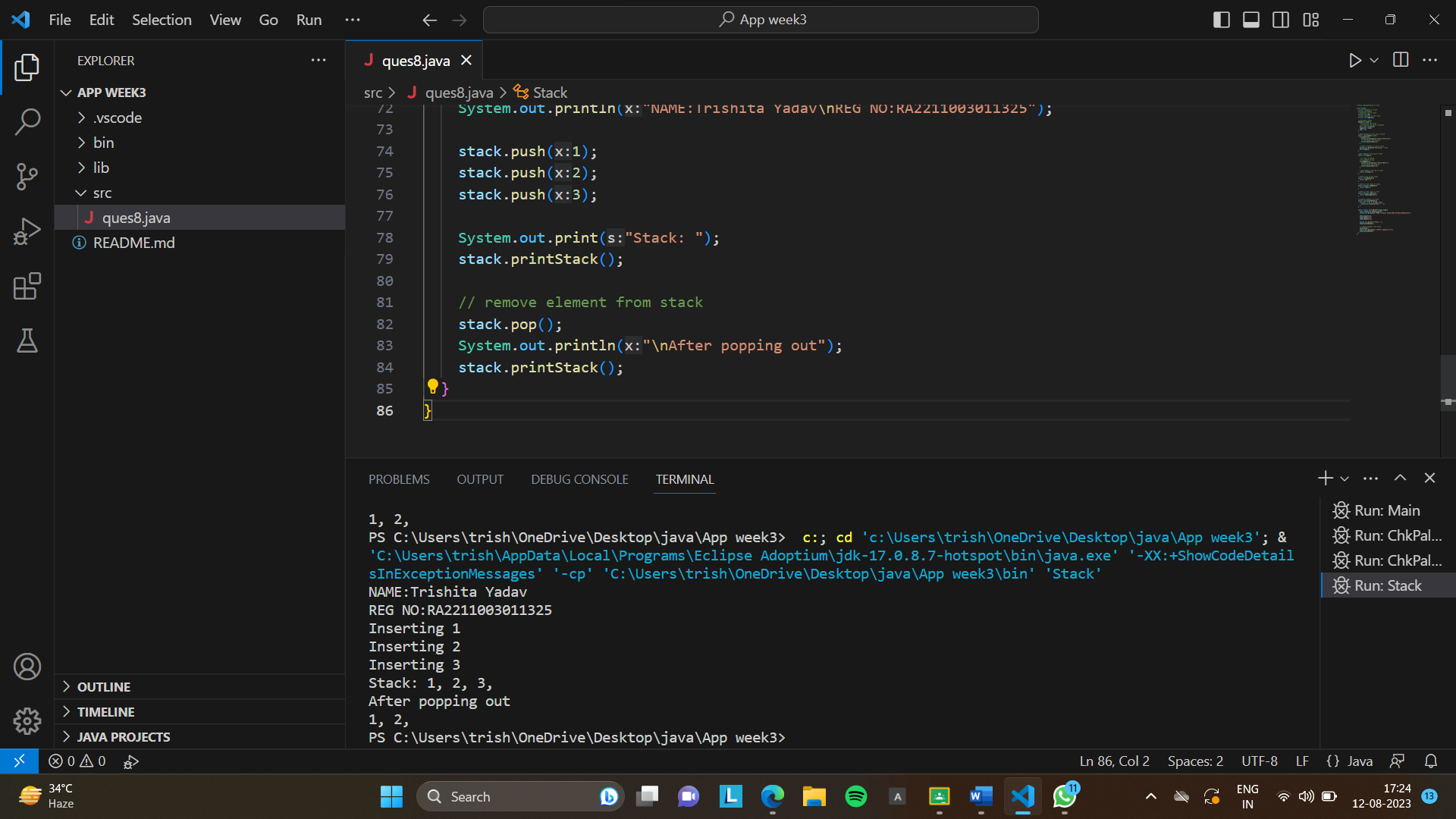
    stack.pop();

    System.out.println("\nAfter popping out");

    stack.printStack();

  }

}

****

**9) Implement a program that uses dynamic memory allocation to simulate a banking system**

**that stores customer information, account details, and transactions.**

#include <iostream>

using namespace std;

// class

class Bank {

private:

int acno;

char name[30];

long balance;

public:

void OpenAccount()

{

cout << "\nEnter Account Number:";

cin >> acno;

cout << "Enter Name:";

cin >> name;

cout << "Enter Balance:";

cin >> balance;

}

void ShowAccount()

{

cout << "Account Number: " << acno << endl;

cout << "Name: " << name << endl;

cout << "Balance: " << balance << endl;

}

void Deposit()

{

long amt;

cout << "Enter Amount U want to deposit? ";

cin >> amt;

balance = balance + amt;

}

void Withdrawal()

{

long amt;

cout << "Enter Amount U want to withdraw? ";

cin >> amt;

if (amt <= balance)

balance = balance - amt;

else

cout << "Less Balance..." << endl;

}

int Search(int);

};

int Bank::Search(int a)

{

if (acno == a) {

ShowAccount();

return (1);

}

return (0);

}

// main code

int main()

{

cout<<"NAME:Trishita Yadav\nREG NO:RA2211003011325";

Bank C[3];

int found = 0, a, ch, i;

for (i = 0; i <= 2; i++) {

C[i].OpenAccount();

}

do {

// display options

cout << "\n\n1:Display All\n2:By Account No\n3:Deposit\n4:Withdraw\n5:Exit" << endl;

// user input

cout << "Please input your choice: ";

cin >> ch;

switch (ch) {

case 1: // displating account info

for (i = 0; i <= 2; i++) {

C[i].ShowAccount();

}

break;

case 2: // searching the record

cout << "Account Number? ";

cin >> a;

for (i = 0; i <= 2; i++) {

found = C[i].Search(a);

if (found)

break;

}

if (!found)

cout << "Record Not Found" << endl;

break;

case 3: // deposit operation

cout << "Account Number To Deposit Amount? ";

cin >> a;

for (i = 0; i <= 2; i++) {

found = C[i].Search(a);

if (found) {

C[i].Deposit();

break;

}

}

if (!found)

cout << "Record Not Found" << endl;

break;

case 4: // withdraw operation

cout << "Account Number To Withdraw Amount? ";

cin >> a;

for (i = 0; i <= 2; i++) {

found = C[i].Search(a);

if (found) {

C[i].Withdrawal();

break;

}

}

if (!found)

cout << "Record Not Found" << endl;

break;

case 5: // exit

cout << "Have a nice day" << endl;

break;

default:

cout << "Wrong Option" << endl;

}

} while (ch != 5);

return 0;

}

