Data Structures

Lab 1

Q1. Get two values from the user and swap them.

#include <iostream>

using namespace std;

int main(){

int a,b;

cout<<"Enter first number ";

cin>>a;

cout<<"Enter second number";

cin>>b;

cout<<"Before Swaping "<<a<<" "<<b<<endl;

int temp = a;

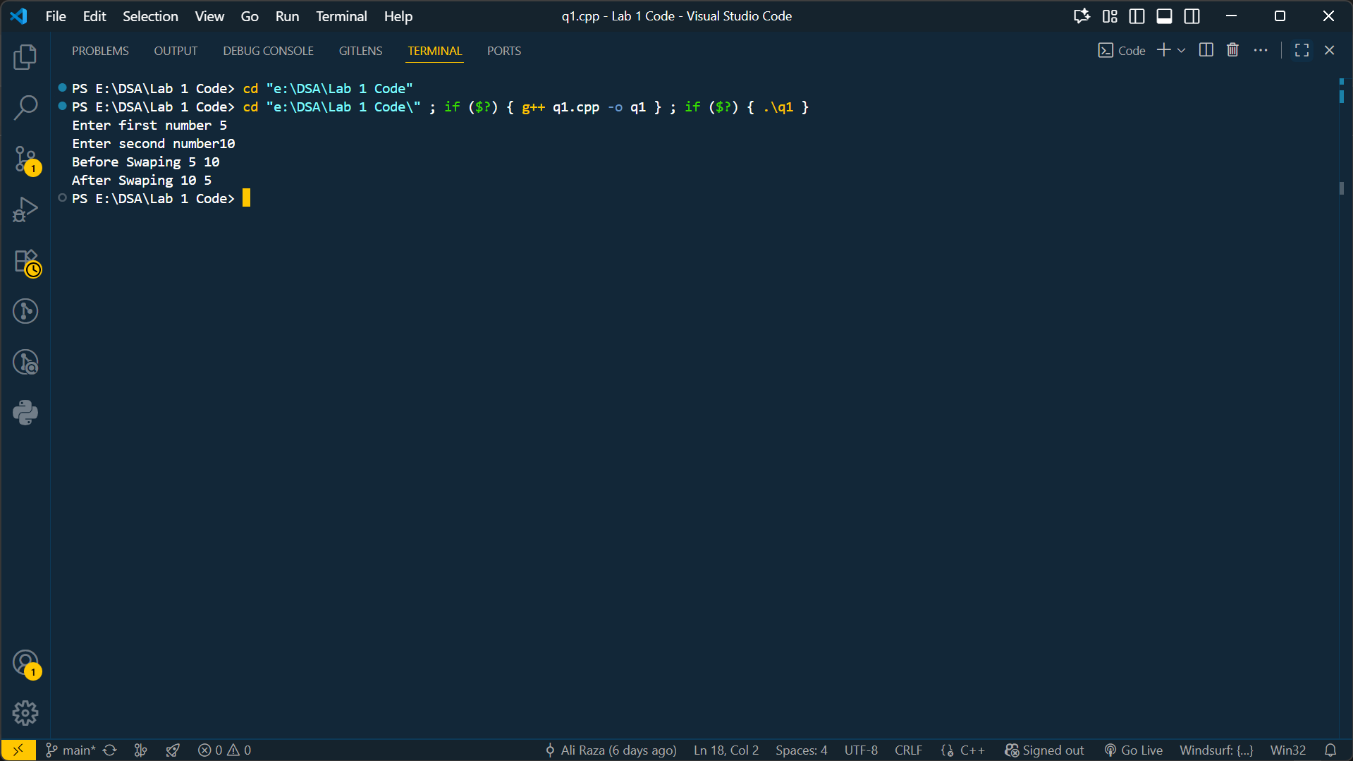
a = b;

b = temp;

cout<<"After Swaping "<<a<<" "<<b;

return 0;

}



Q2 . Ask user to enter a three digit number. Then display the number in reverse order.

#include <iostream>

using namespace std;

int main(){

int a;

cout<<"Enter a three digit Number\t";

cin>>a;

while (a>0)

{

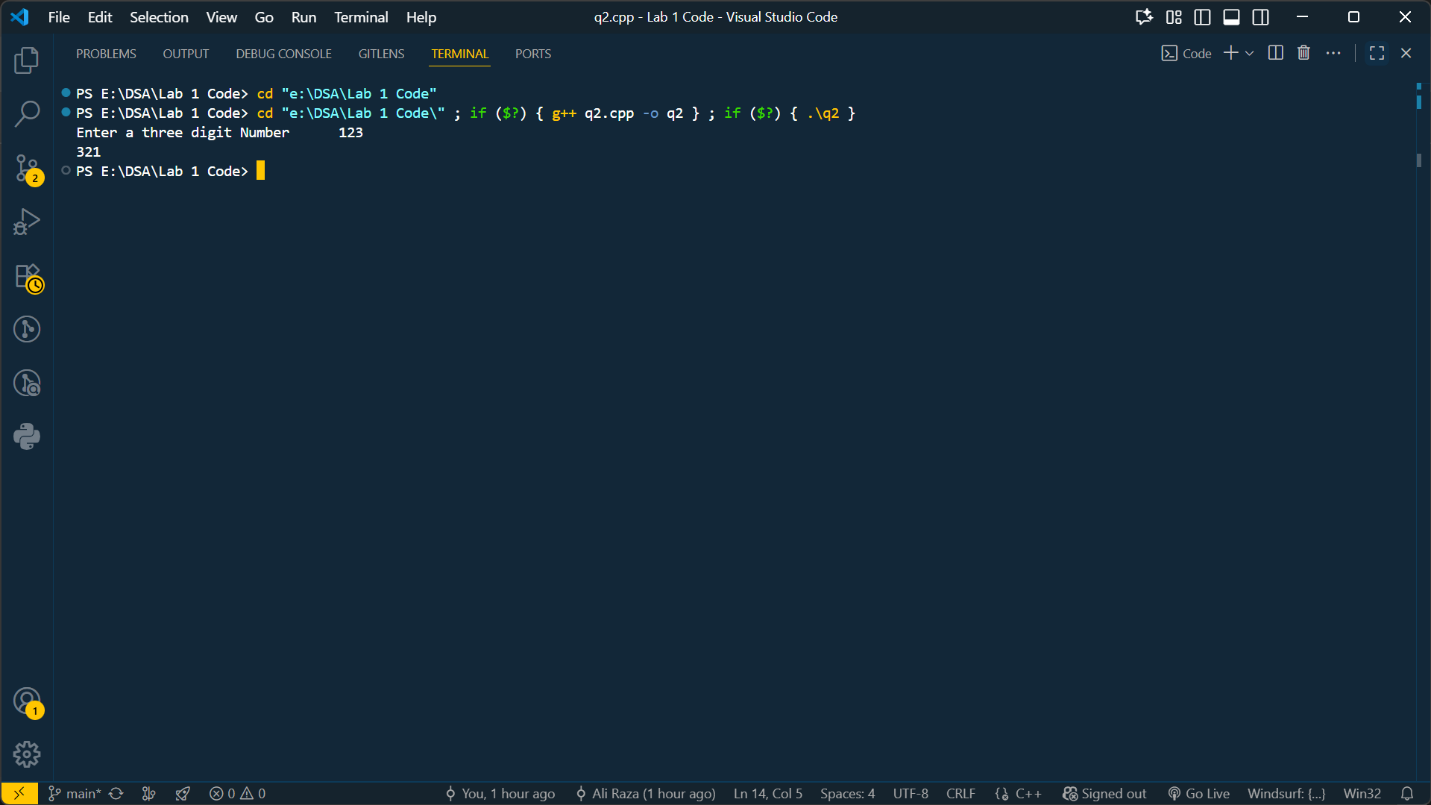
cout<<a%10;

a/=10;

}

return 0;

}



Q3. A program that takes an n digits integer from user and shows the digits on the screen separately i.e. if user enters 6572, it displays 6,5,7,2 separately and a total of individual numbers as well e.g 20 in given case

#include <iostream>

using namespace std;

int main(){

string num ;

cout<<"Enter a digit ";

getline(cin,num);

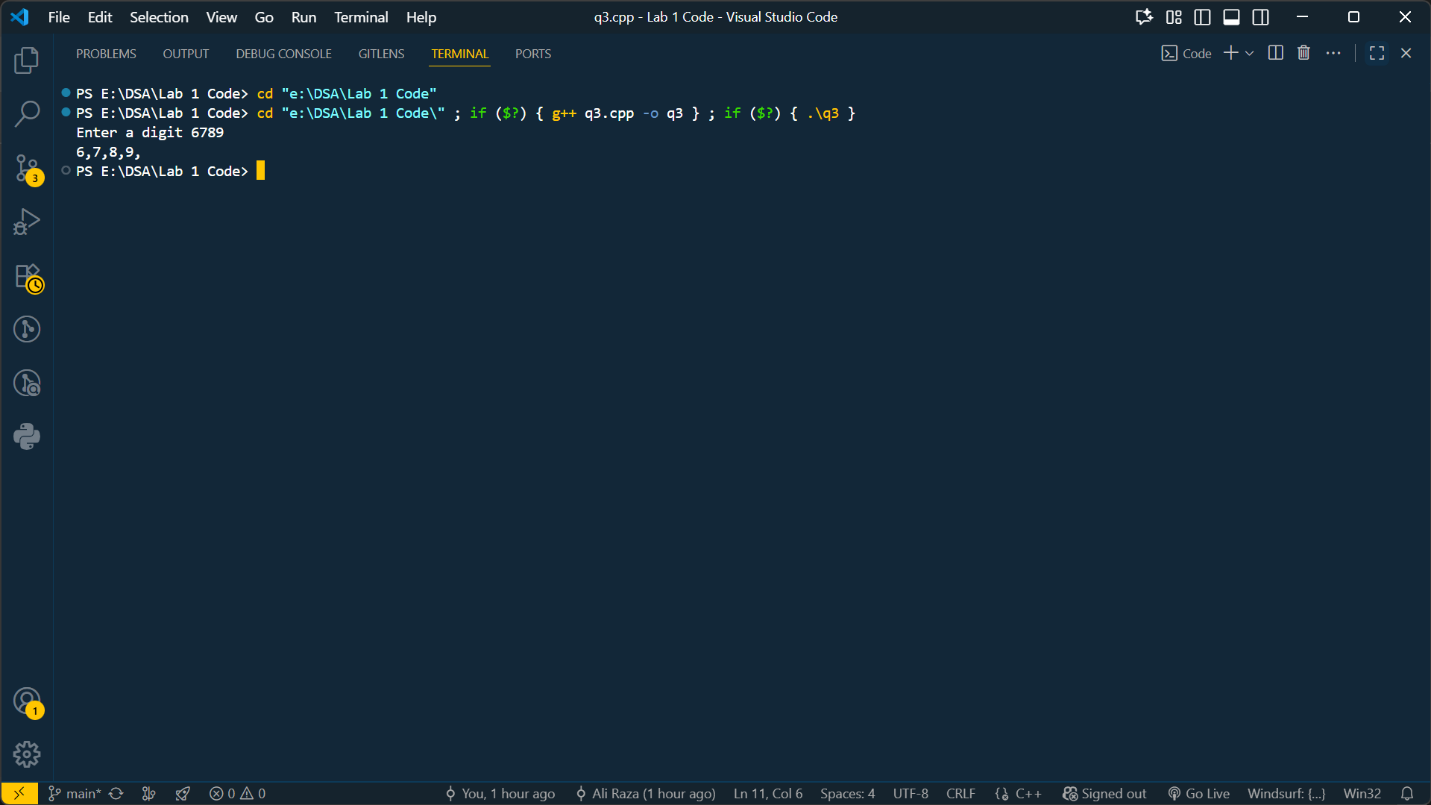
for(char c:num){

cout<<c<<",";

}

return 0;

}



Q4. Write a program that takes radius of a circle from the user and calculates the diameter, circumference and area of the circle and display the result.

#include <iostream>

using namespace std;

int main(){

float r,dia,circum,area;

cout<<"Enter Radius of Circle\t";

cin>>r;

dia = r\*2;

cout<<"Diameter of Circle is \t"<<dia<<endl;

circum = 2\*3.14\*r;

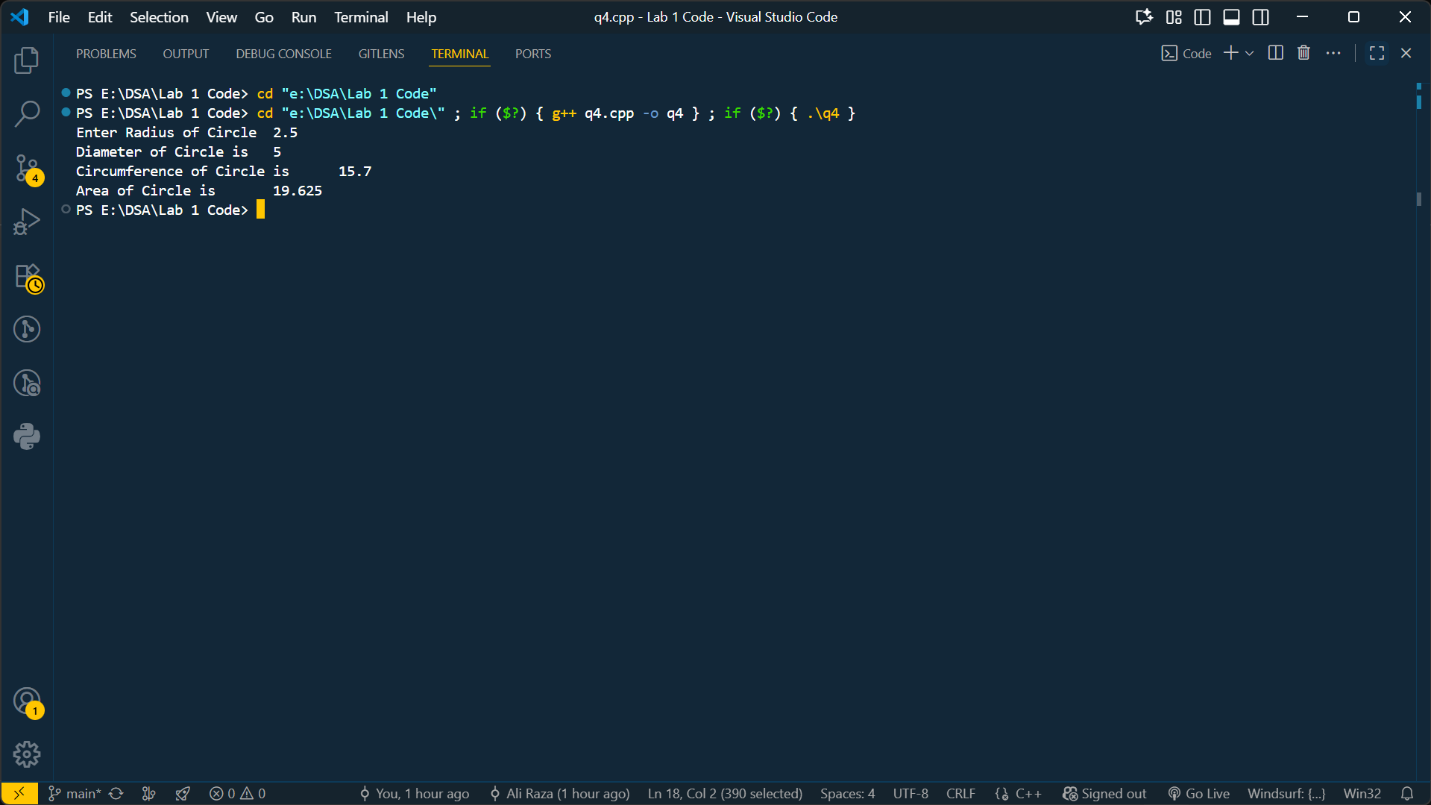
cout<<"Circumference of Circle is \t"<<circum<<endl;

area = 3.14 \*r \*r ;

cout<<"Area of Circle is \t"<<area<<endl;

return 0;

}



Q5 . Write a program which calculates and displays the sum of first 100 integers.

*#include* <iostream>

*using* *namespace* std;

int main(){

    int limit *=* 100;

    int sum *=* 0;

*for* (int i *=* 1; i *<=*100 ; i*++*)

    {

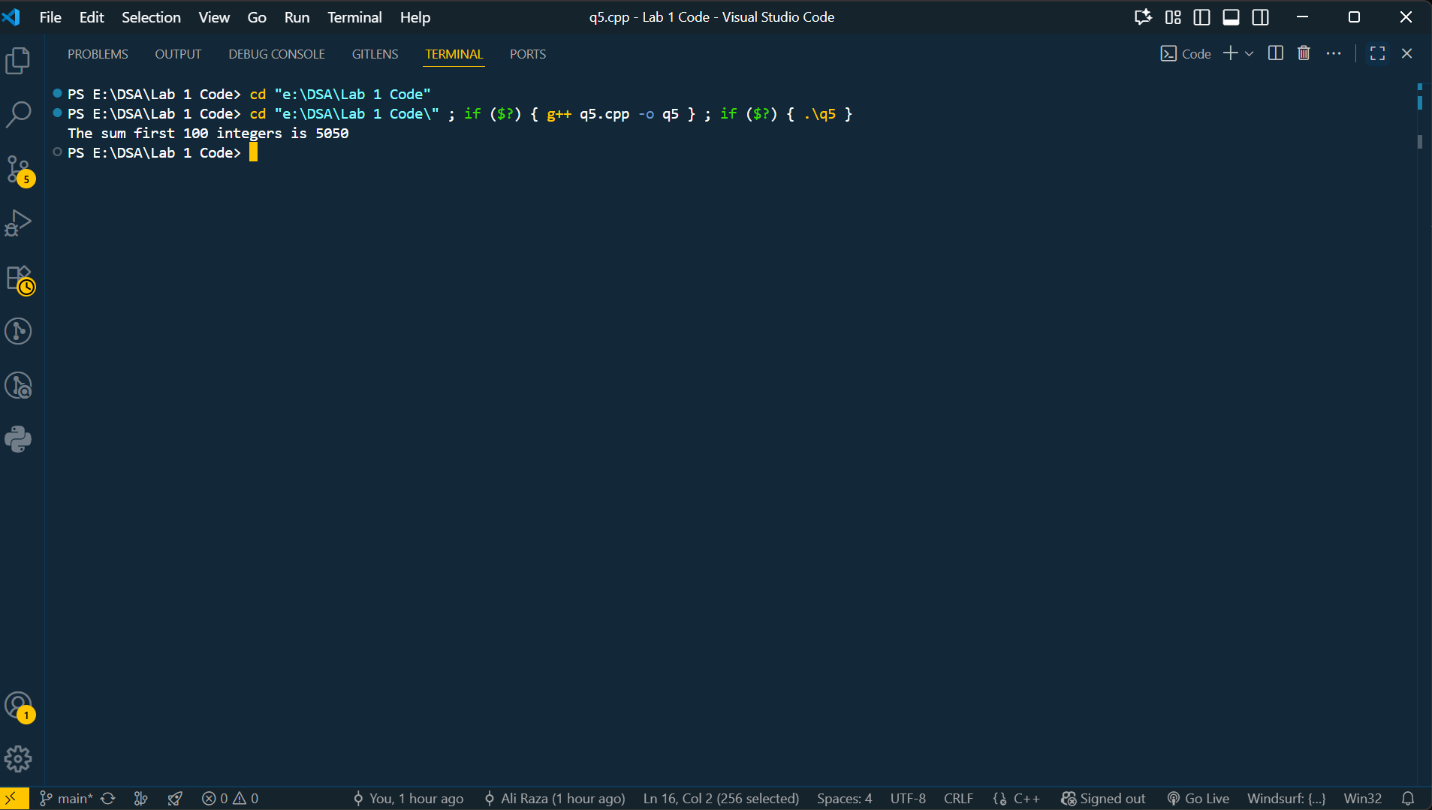
        sum*+=*i;

    }

    cout*<<*"The sum first "*<<*limit*<<*" integers is "*<<*sum*<<*endl;

*return* 0;

}



Q6. Write a program that calculates sum of even numbers for a given upper limit of integers. The user should not be able to give upper limit greater than 1000. The program should contain two functions. The first function GetUpperLimit takes the input from the user and second function SumOfEven calculates the sum of even number up to given upper limit.

*#include* <iostream>

*using* *namespace* std;

int getUpperLimit(){

    int limit;

    cout*<<*"Enter the limit\t";

    cin*>>*limit;

*if* (limit*>*1000)

    {

        cout*<<*"Please enter valid Limit\n";

*return* 0;

    }*else*{

*return* limit;

    }

}

int sumOfEven(int num){

    int sum *=* 0;

*for* (int i *=* 0; i *<=*num; i*+=*2)

    {

        sum*+=*i;

    }

*return* sum;

}

int main(){

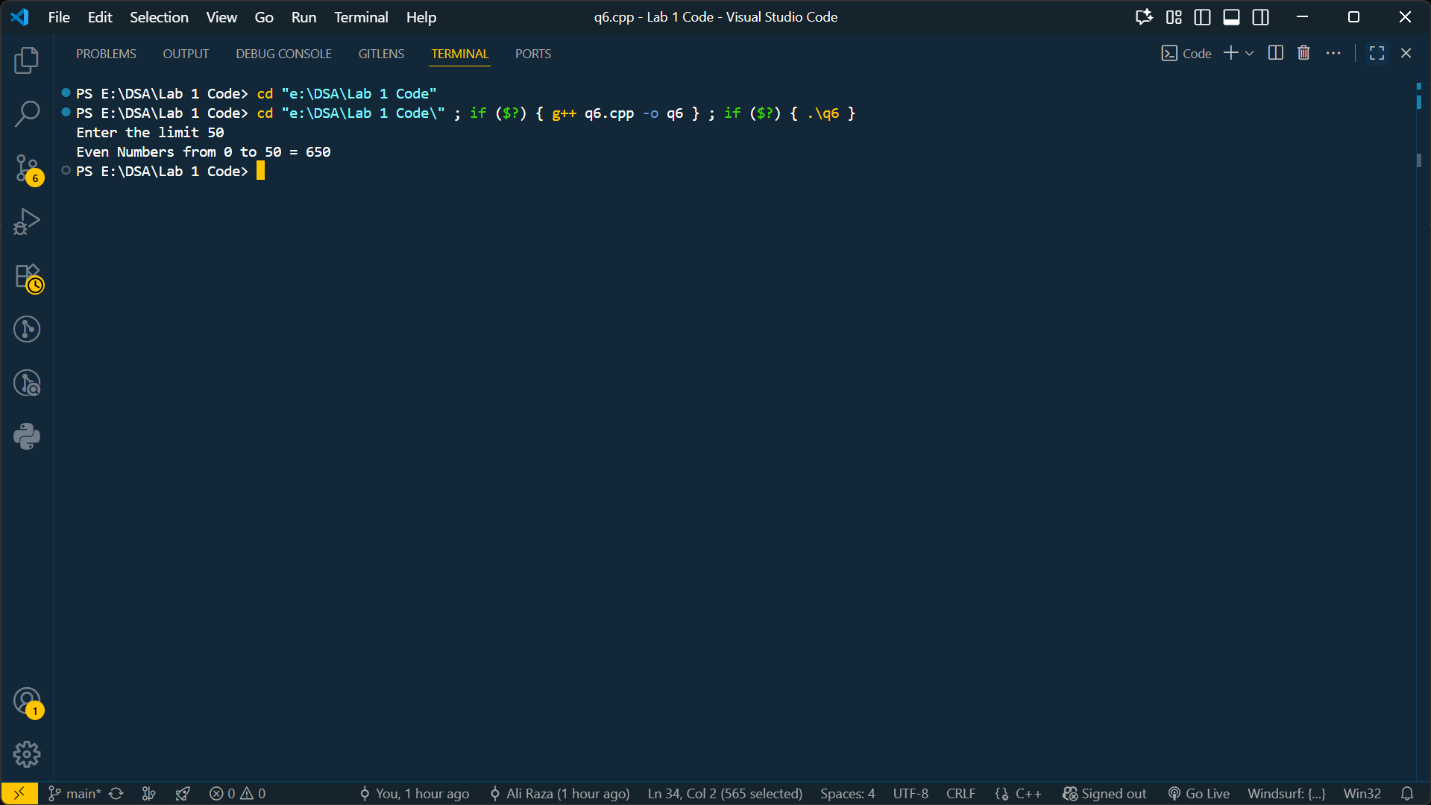
    int num *=* getUpperLimit();

    int sum *=* sumOfEven(num);

    cout*<<*"Even Numbers from 0 to "*<<*num*<<*" = "*<<*sum;

*return* 0;

}



Q7. Write a function that calculates the factorial of a given number by using iteration and recursion

*#include* <iostream>

*using* *namespace* std;

int facByIter(int n){

    int fac *=* 1;

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

    {

        fac *\*=* i;

    }

*return* fac;

}

int facByRec(int n){

*if* (n *==* 1)

    {

*return* 1;

    }*else*{

*return* n*\** facByRec(n*-*1);

    }

}

int main(){

    int num *=* 5;

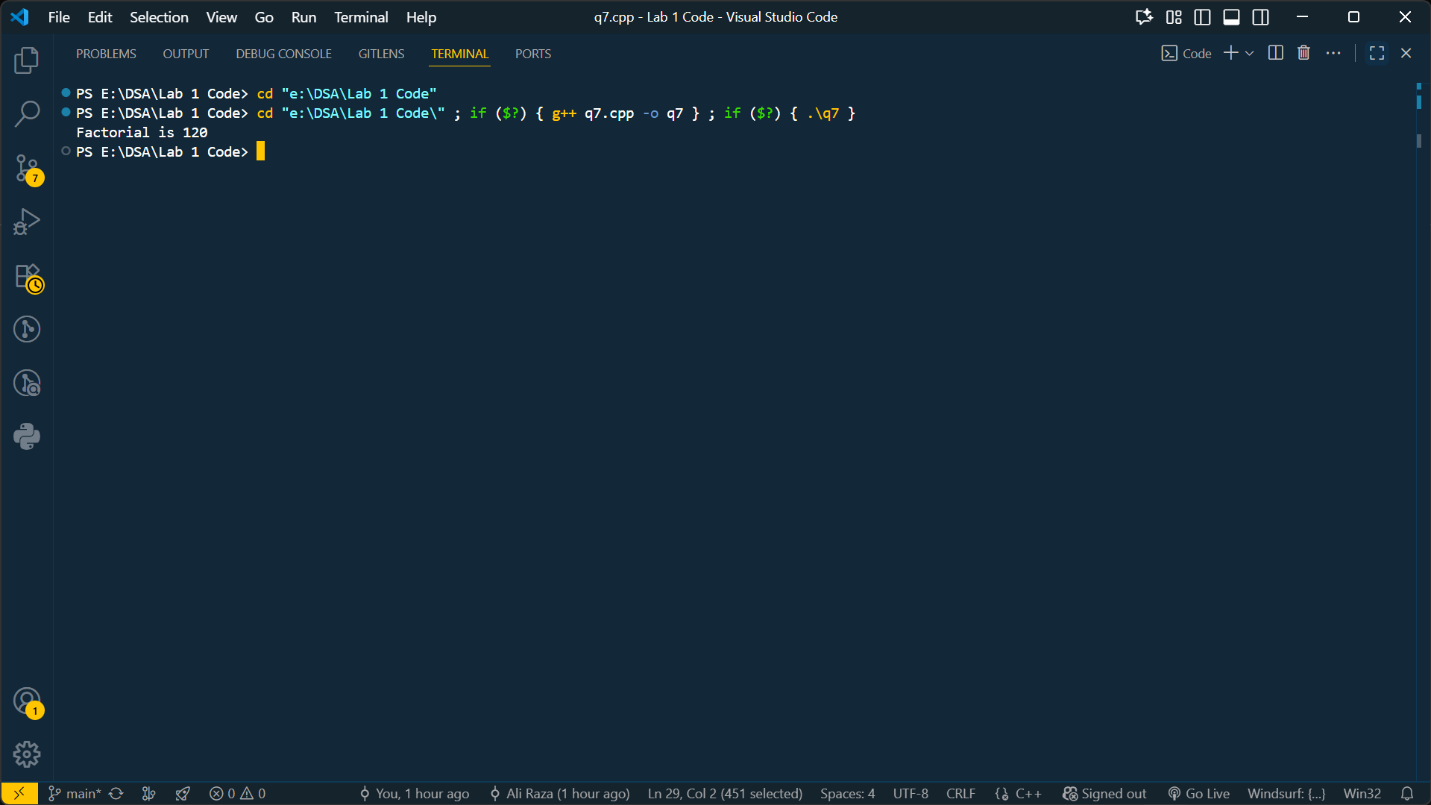
*// int fac = facByIter(num);*

    int fac *=* facByRec(num);

    cout*<<*"Factorial is "*<<*fac;

*return* 0;

}



Q8. Write a program which takes an integer input from user and displays its table. The table is displayed up to the multiplier entered by the user

*#include* <iostream>

*using* *namespace* std;

void tablePrint(int num,int multiplier){

*for* (int i *=* 1; i *<=* multiplier; i*++*)

    {

        cout*<<*num*<<*" \* "*<<*i*<<*" = "*<<*num*\**i*<<*endl;

    }

}

int main(){

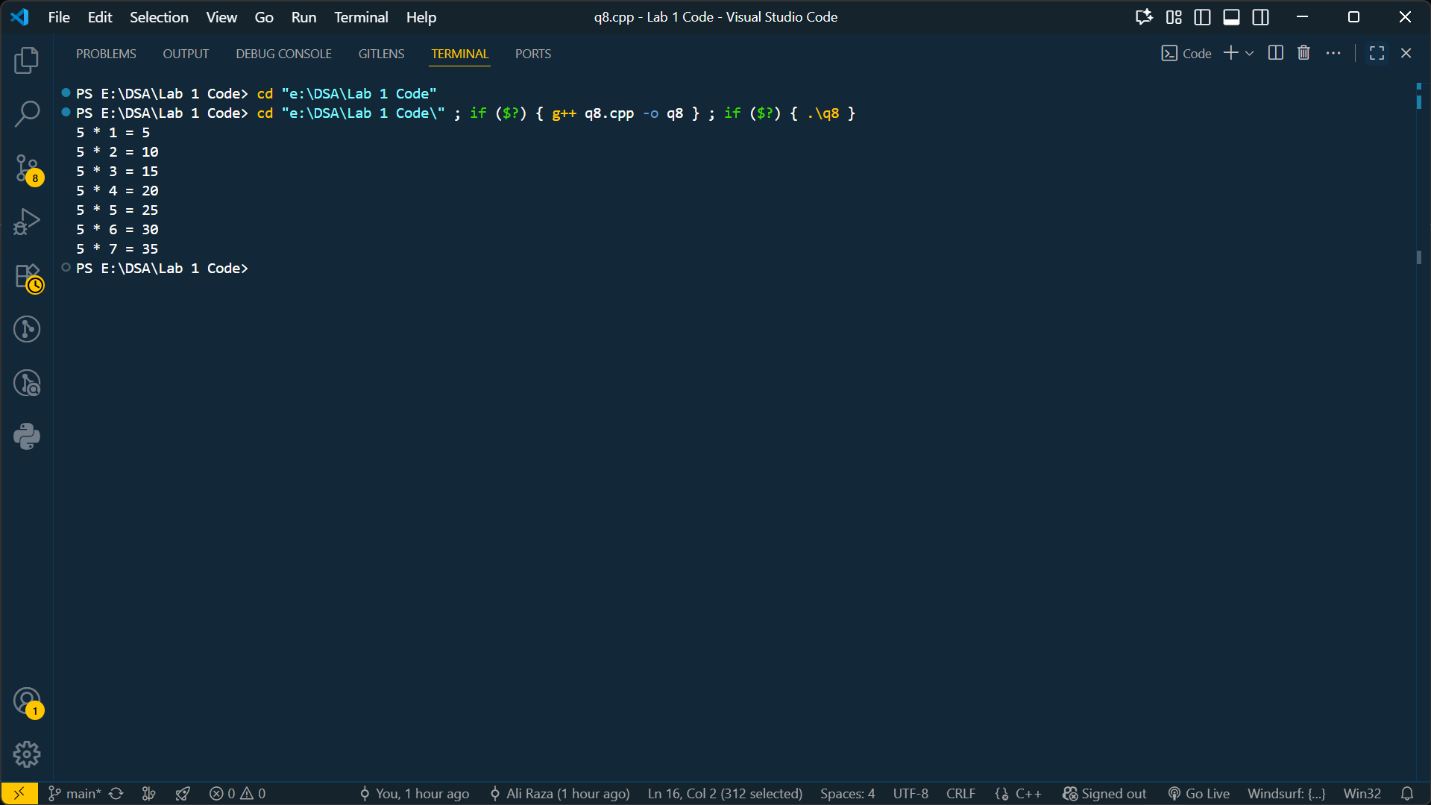
    int table *=* 5;

    int multiplier *=* 7;

    tablePrint(table,multiplier);

*return* 0;

}



Q9. Write a program that allows the user to guess a character from ‘a’ to ‘z’. The user should allow maximum five tries for guessing. If the user guesses the number on the first try then he/she gets score 10,000. On second, third, fourth and fifth try gets 8,000, 6,000, 4,000, 2,000 respectively and 0 if he/she could not guess the number.

*#include* <iostream>

*#include* <cstdlib>

*#include* <ctime>

*using* *namespace* std;

int main(){

    int tries *=* 1;

    srand(time(0));

    char randomChar *=* 'a' *+* rand() *%* 26;

*// cout<<"The random Character is "<<randomChar<<endl;*

    char c;

    int score *=* 0;

*do*

    {

        cout*<<*"Enter any character\t";

        cin*>>*c;

*if*(c *==* randomChar){

*if* (tries *==* 1)

            {

                score *+=*10000;

            }*else* *if*(tries *==* 2){

                score*+=*8000;

            }

*else* *if*(tries *==* 3){

                score*+=*6000;

            }

*else* *if*(tries *==* 4){

                score*+=*4000;

            }

*else* *if*(tries *==* 5){

                score*+=*2000;

            }

*break*;

        }

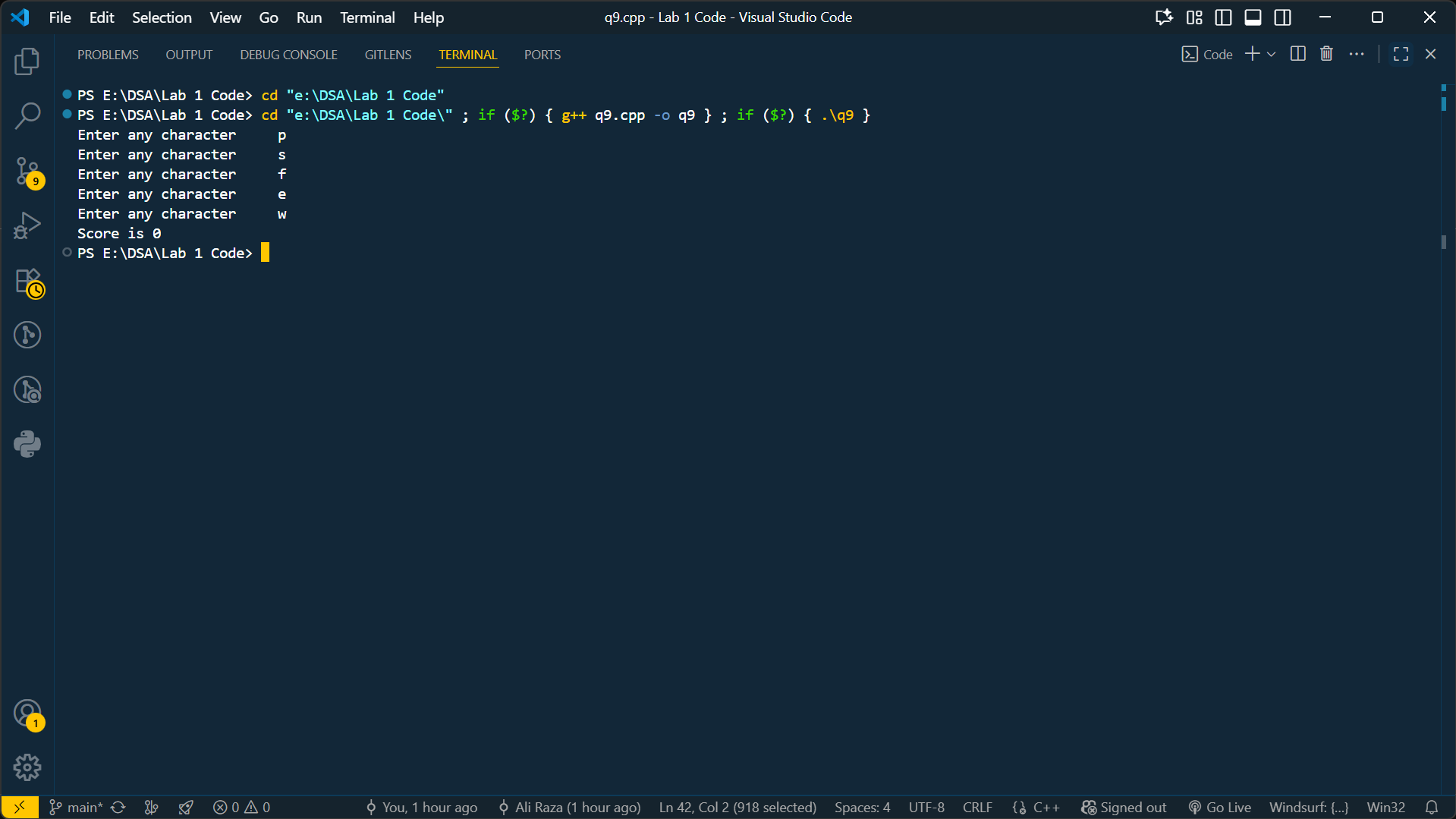
        tries*++*;

    } *while* (tries*<=*5);

    cout*<<*"Score is "*<<*score;

*return* 0;

}



Q10. Write a program to display string from backward e.g. if user enters “Hello World” the program should display “ dlroW olleH”.

*#include* <iostream>

*using* *namespace* std;

int main(){

    string s;

    cout*<<*"Enter a string\t";

    getline(cin,s);

*for* (int i *=* s.length(); i *>=* 0; i*--*)

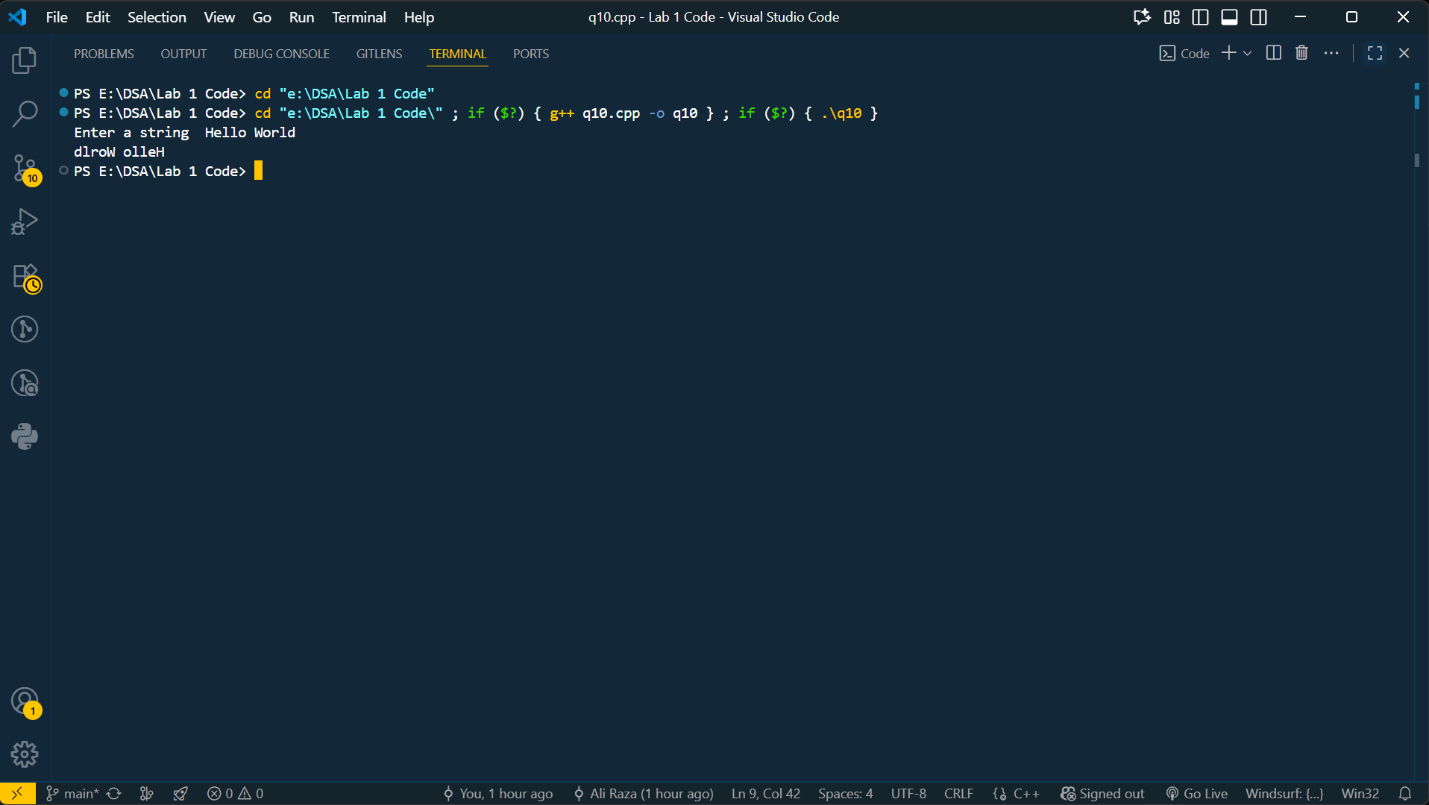
    {

        cout*<<*s*[*i*]*;

    }

*return* 0;

}



Q11. Write a program to count number of words in string e.g. if user enters “This is a string” the program should display 4.

*#include* <iostream>

*using* *namespace* std;

int main(){

    int spaces *=* 1;

    string s;

    cout*<<*"Enter a string\t";

    getline(cin,s);

*for*(char c:s){

*if* (c*==*' ')

        {

            spaces*++*;

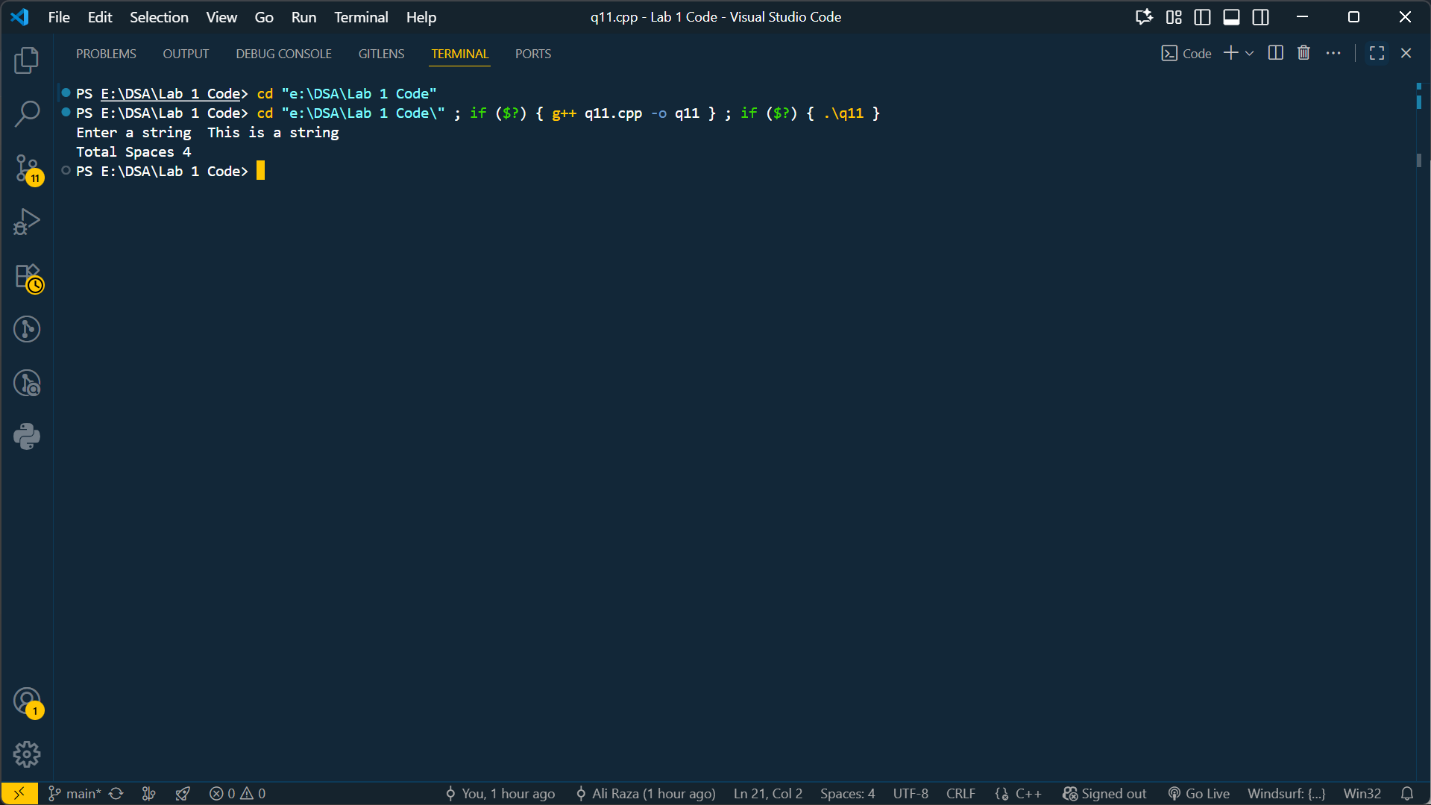
        }

    }

    cout*<<*"Total Spaces "*<<*spaces*<<*endl;

*return* 0;

}

****

Lab 2

Write a program which calculates the area of a shape. The program should consist of two functions: a) The first function names GetChoice should take the choice of input. Ask the user to input the shape for which area has to be calculated. The shapes are Square, Rectangle and Circle. b) The second function named CalculateArea calculates the area according to the user choice.

#include <iostream>

using namespace std;

void getChoice(char &ch){

cout<<"Enter Your Choice(s,r,c)";

cin>>ch;

}

float squareArea(float len){

return len\*len;

}

float rectangleArea(float len,float wid){

return len\*wid;

}

float circleArea(float r){

return 3.14 \* r \*r;

}

void CalculateArea(char choice){

float area;

if (choice == 's')

{

float len;

cout<<"Enter the length";

cin>>len;

area = squareArea(len);

cout<<"Area is "<<area;

}

if (choice == 'r')

{

float len,wid;

cout<<"Enter the length";

cin>>len;

cout<<"Enter width";

cin>>wid;

area = rectangleArea(len,wid);

cout<<"Area is "<<area;

}

if (choice == 'c')

{

float r;

cout<<"Enter the radius";

cin>>r;

area = circleArea(r);

cout<<"Area is "<<area;

}

}

int main(){

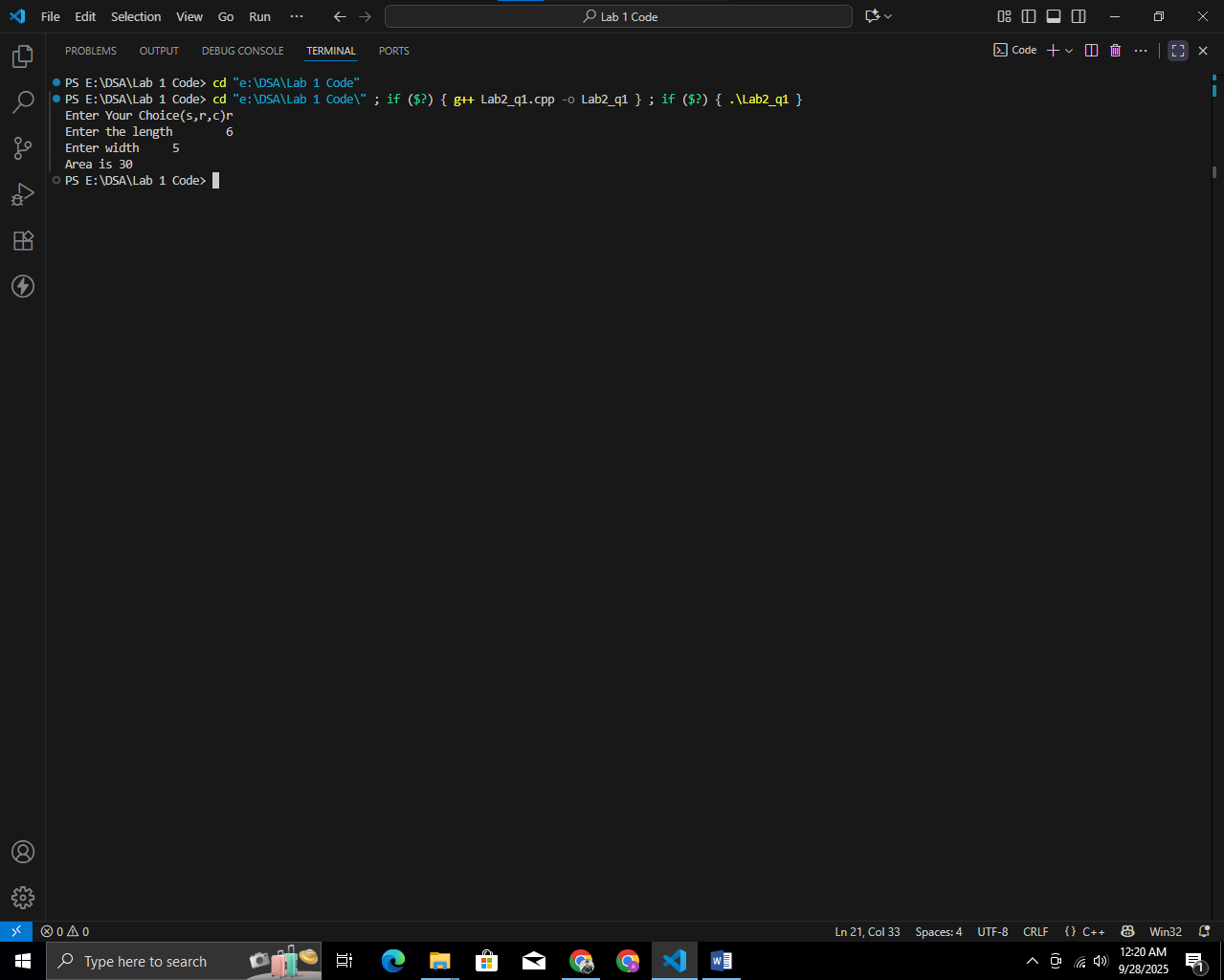
char shape;

getChoice(shape);

CalculateArea(shape);

return 0;

}



Q2. A List stores details of 25 students (rollno, name, and marks in three subjects). Write a program to create such a list and print out a list of students who have failed in more than one subject.

#include <iostream>

#include <string>

using namespace std;

class Student {

private:

int rollNo;

string name;

int marks[3];

public:

void inputData() {

cout << "Enter Roll No: ";

cin >> rollNo;

cin.ignore();

cout << "Enter Name: ";

getline(cin, name);

cout << "Enter marks in 3 subjects: ";

for (int i = 0; i < 3; i++) {

cin >> marks[i];

}

}

int failed() {

int failCount = 0;

for (int i = 0; i < 3; i++) {

if (marks[i] < 40) {

failCount++;

}

}

return failCount;

}

void display() {

cout << "\nRoll No: " << rollNo;

cout << "\nName: " << name;

cout << "\nMarks: ";

for (int i = 0; i < 3; i++) {

cout << marks[i] << " ";

}

cout << "\n";

}

};

int main() {

Student students[3];

cout << "Enter details of 3 students:\n";

for (int i = 0; i < 3; i++) {

cout << "\nStudent " << (i + 1) << ":\n";

students[i].inputData();

}

cout << "\nStudents who failed in more than one subject:\n";

for (int i = 0; i < 3; i++) {

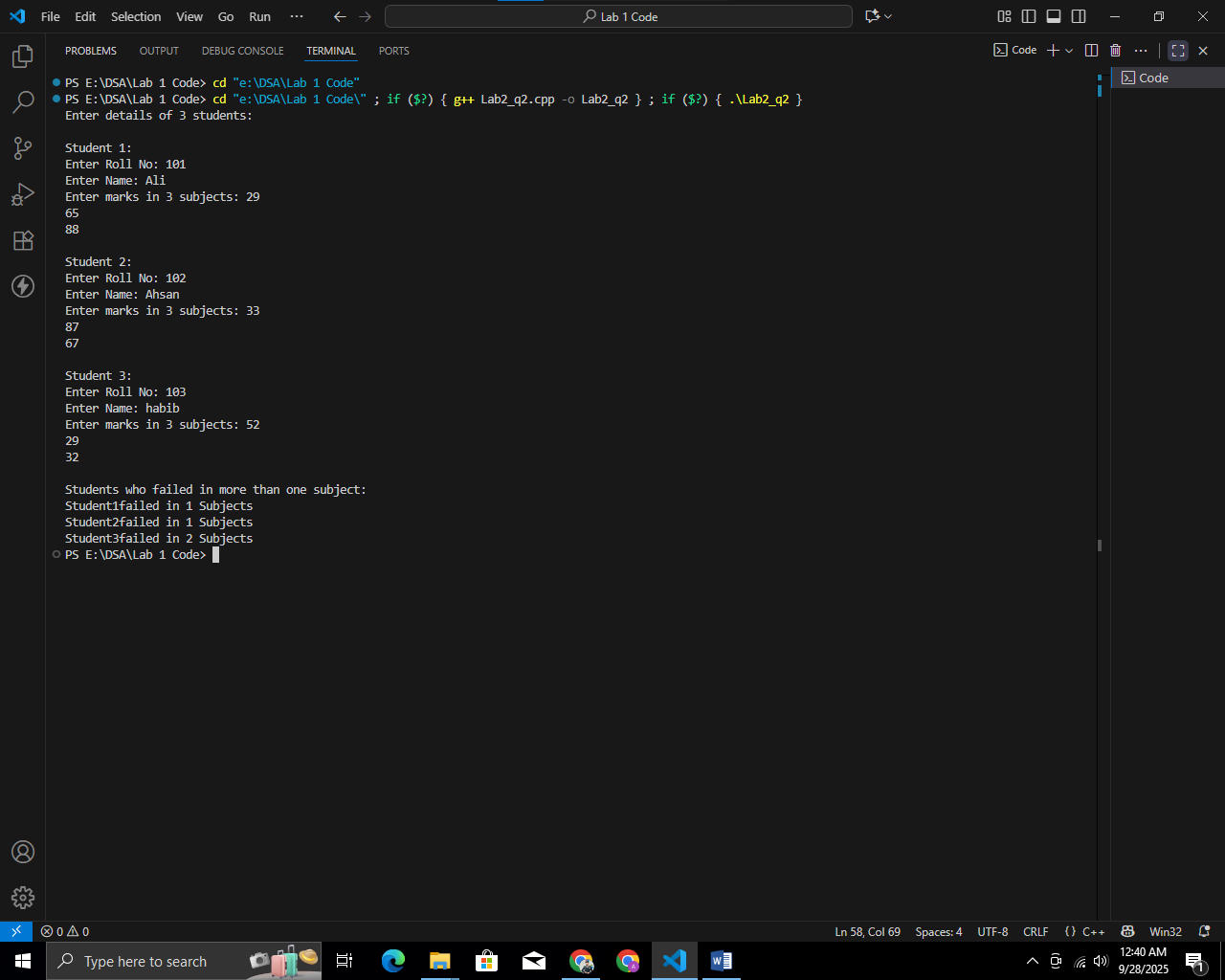
int fail = students[i].failed();

cout<<"Student"<<i+1<<"failed in "<<fail<<" Subjects"<<endl;

}

return 0;

}



Q3. Write a program which takes 10 integers from the user and sort them in ascending order. Sort the array elements using a function AscendingSort which sort the elements using Bubble Sort. The function takes the array as argument sorts them

#include <iostream>

using namespace std;

void AscendingSort(int arr[], int n) {

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

int main() {

const int SIZE = 10;

int numbers[SIZE];

cout << "Enter 10 integers: ";

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

cin >> numbers[i];

}

AscendingSort(numbers, SIZE);

cout << "\nNumbers in ascending order: ";

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

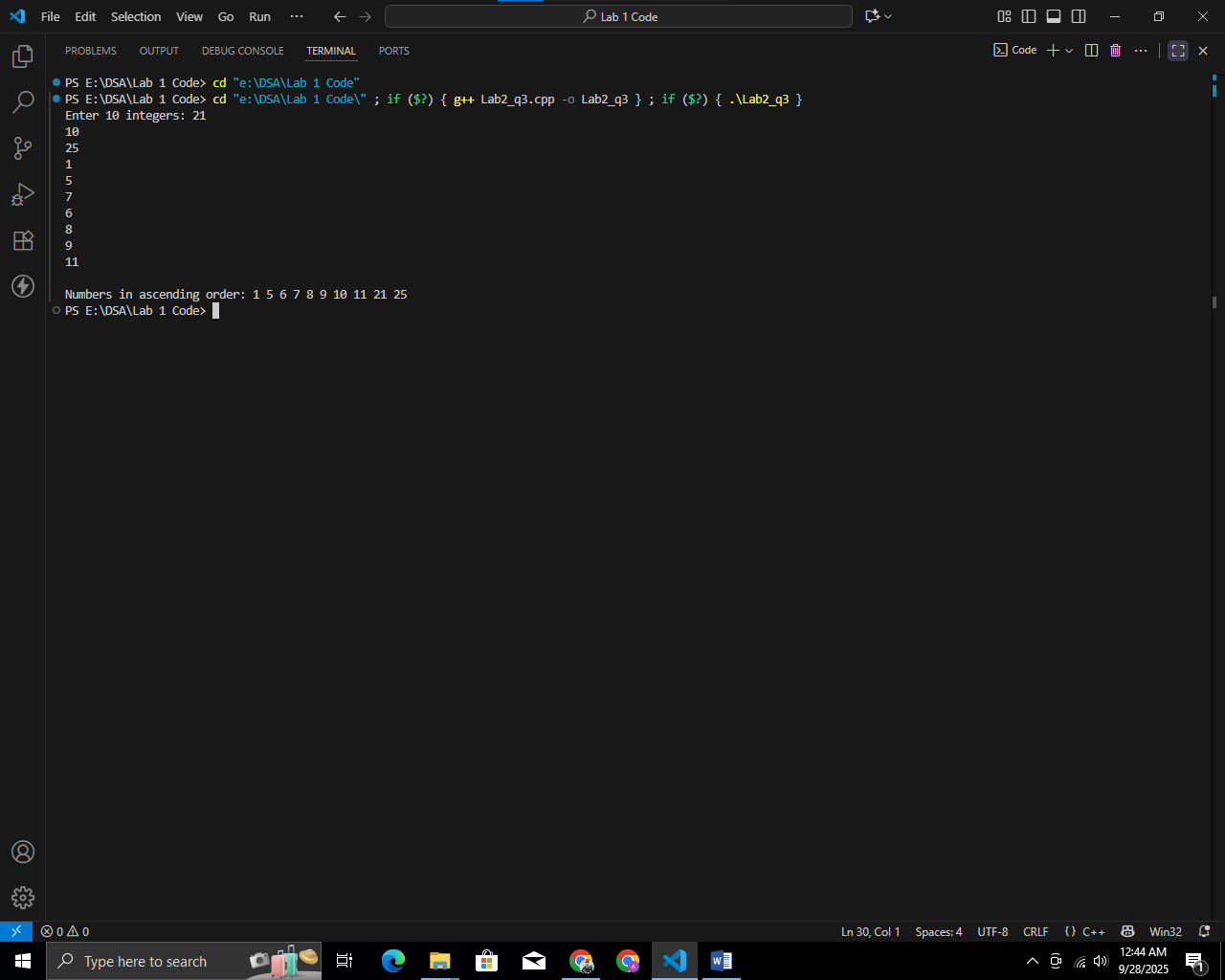
cout << numbers[i] << " ";

}

cout << endl;

return 0;

}



Q4. Write a class Person which has name, age and CNIC number data member. Provide a parameterized constructor of the class and a member function Print() which will display the data members of the class.

#include <iostream>

#include <string>

using namespace std;

class Person {

private:

string name;

int age;

string cnic;

public:

Person(string n, int a, string c) {

name = n;

age = a;

cnic = c;

}

void Print() {

cout << "\nName: " << name;

cout << "\nAge: " << age;

cout << "\nCNIC: " << cnic << endl;

}

};

int main() {

string n, cnic;

int a;

cout << "Enter Name: ";

getline(cin, n);

cout << "Enter Age: ";

cin >> a;

cin.ignore();

cout << "Enter CNIC number: ";

getline(cin, cnic);

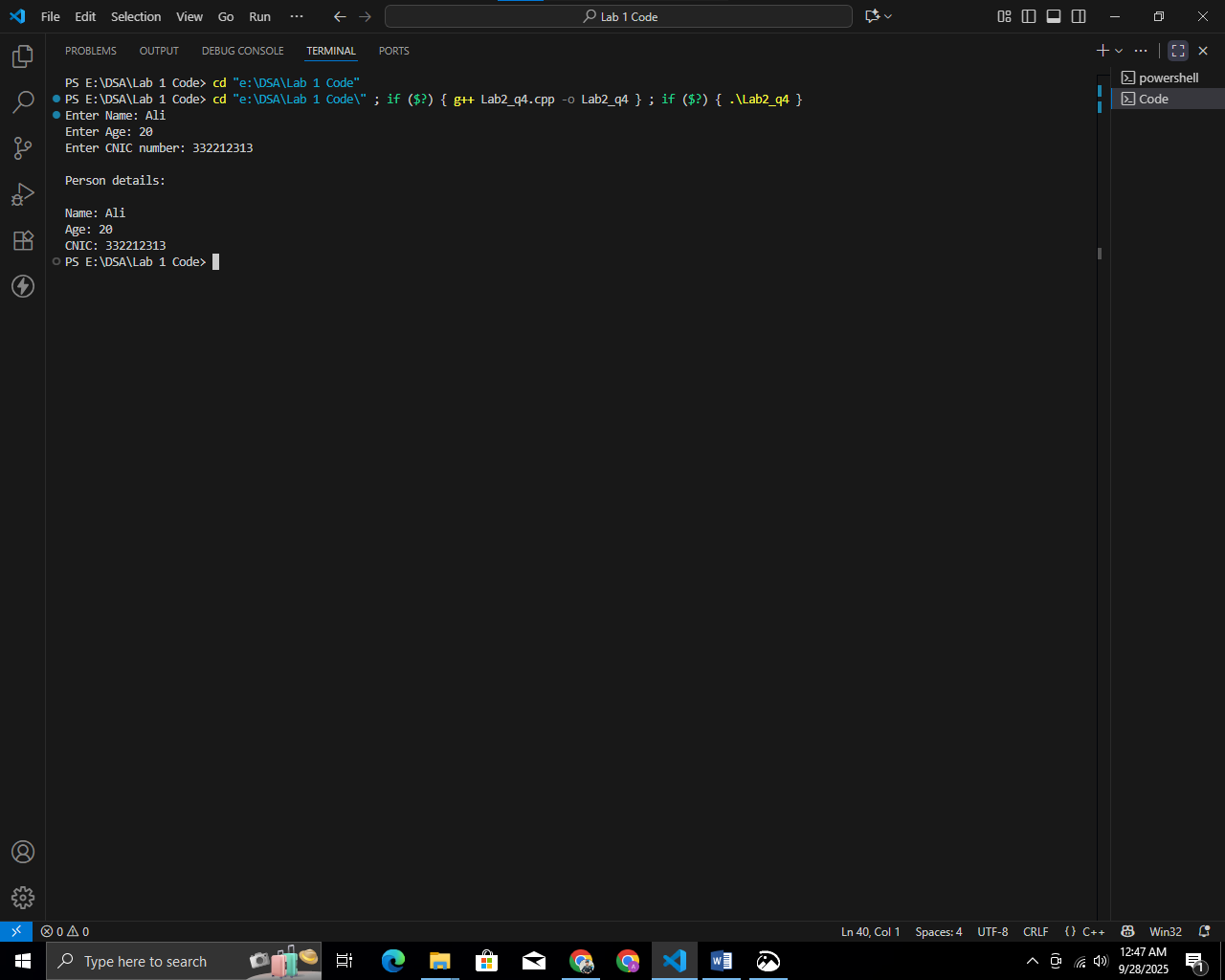
Person p1(n, a, cnic);

cout << "\nPerson details:\n";

p1.Print();

return 0;

}



Q5. Edit the previously built class Person and write getter and setter functions despite of parameterized constructor of the class. The class should not be allow to set age less than 0 years and more than more than 130 years. Similarly the class does not allow to set CNIC number whose length more than or less than 13(e.g. 3312059087658) digits.

#include <iostream>

#include <string>

using namespace std;

class Person {

private:

string name;

int age;

string cnic;

public:

Person() {

name = "";

age = 0;

cnic = "";

}

void setName(const string &n) {

name = n;

}

bool setAge(int a) {

if (a >= 0 && a <= 130) {

age = a;

return true;

} else {

cout << "Invalid age! Must be between 0 and 130.\n";

return false; // invalid

}

}

bool setCNIC(const string &c) {

if (c.length() == 13) {

cnic = c;

return true;

} else {

cout << "Invalid CNIC! Must be exactly 13 digits.\n";

return false;

}

}

string getName() const {

return name;

}

int getAge() const {

return age;

}

string getCNIC() const {

return cnic;

}

void Print() const {

cout << "\nName: " << name;

cout << "\nAge: " << age;

cout << "\nCNIC: " << cnic << endl;

}

};

int main() {

Person p;

string n, c;

int a;

cout << "Enter Name: ";

getline(cin, n);

p.setName(n);

cout << "Enter Age: ";

cin >> a;

while (!p.setAge(a)) {

cout << "Re-enter Age: ";

cin >> a;

}

cin.ignore();

cout << "Enter CNIC number (13 digits): ";

getline(cin, c);

while (!p.setCNIC(c)) {

cout << "Re-enter CNIC: ";

getline(cin, c);

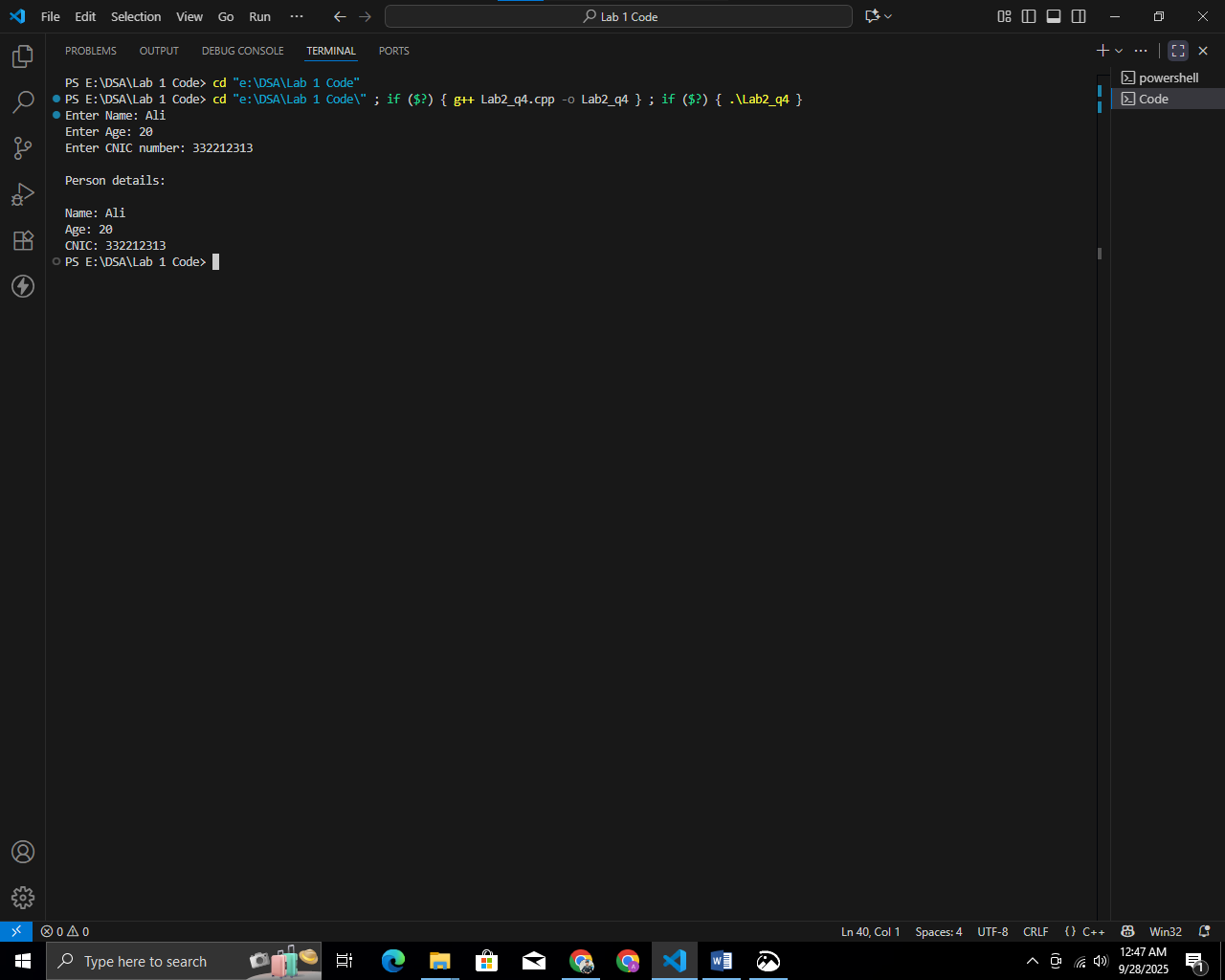
}

cout << "\nPerson details:\n";

p.Print();

return 0;

}



Q6. Write the definition for a class called Distance that has data member feet as integer and inches as float. The class has the following member functions: void set(int, float) to give value to object void disp() to display distance in feet and inches Distance add(Distance) to sum two distances & return distance 1. Write the definitions for each of the above member functions. 2. Write main function to create three Distance objects. Set the value in two objects and call add() to calculate sum and assign it in third object. Display all distances.

#include <iostream>

using namespace std;

class Distance {

private:

int feet;

float inches;

public:

void set(int f, float i) {

feet = f;

inches = i;

if (inches >= 12.0) {

feet += inches / 12;

inches = static\_cast<int>(inches) % 12 + (inches - inches);

}

}

void disp() {

cout << feet << " feet " << inches << " inches" << endl;

}

Distance add(Distance d) {

Distance temp;

temp.inches = inches + d.inches;

temp.feet = feet + d.feet;

if (temp.inches >= 12.0) {

temp.feet += temp.inches / 12;

temp.inches = static\_cast<int>(temp.inches) % 12 + (temp.inches - temp.inches);

}

return temp;

}

};

int main() {

Distance d1, d2, d3;

d1.set(5, 8.5);

d2.set(3, 11.75);

d3 = d1.add(d2);

cout << "Distance 1: ";

d1.disp();

cout << "Distance 2: ";

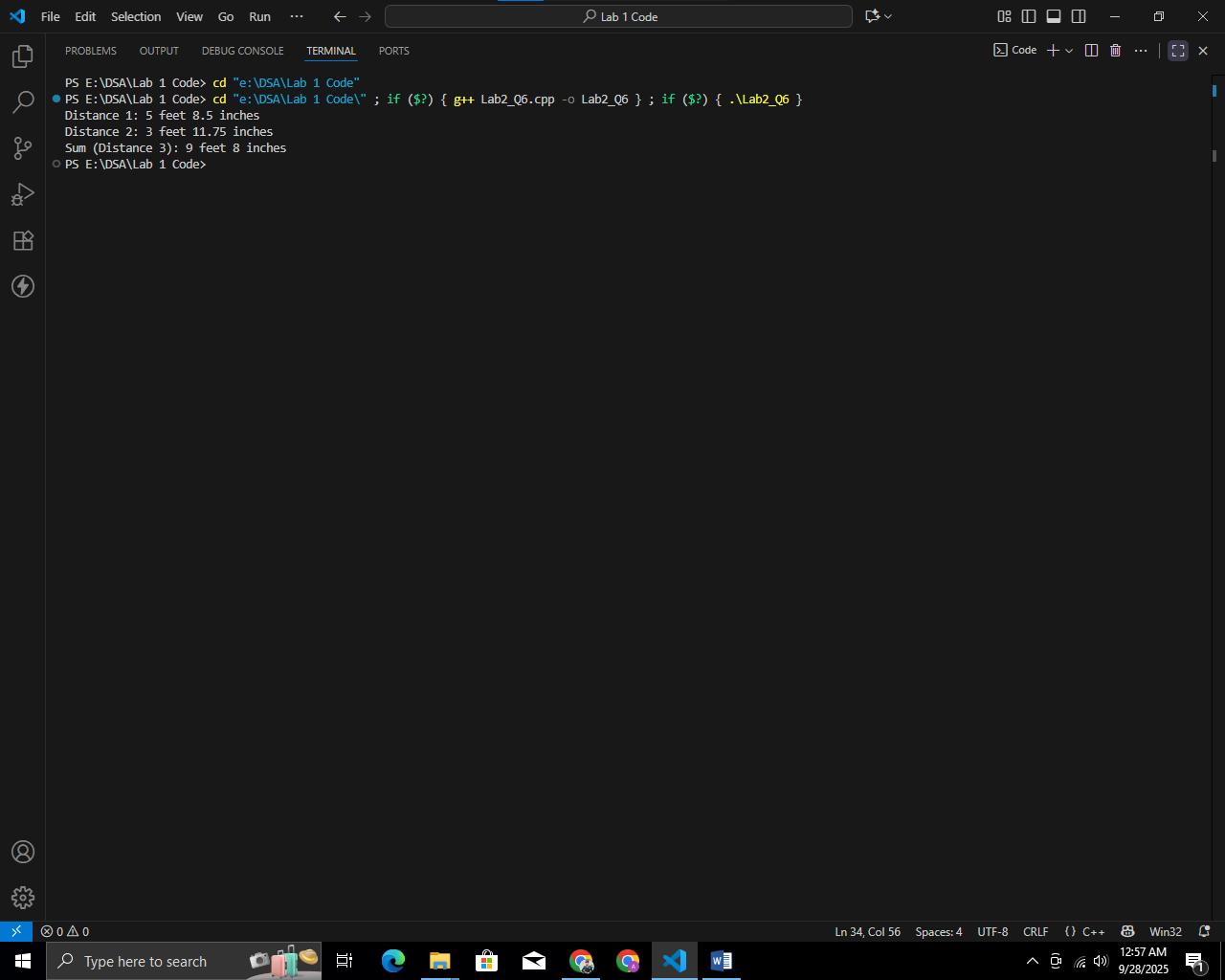
d2.disp();

cout << "Sum (Distance 3): ";

d3.disp();

return 0;

}



Q7. Write a function countEven(int\*, int) which receives an integer array and its size, and returns the number of even numbers in the array.

#include <iostream>

using namespace std;

int countEven(int\* arr, int size) {

int count = 0;

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

if (arr[i] % 2 == 0) {

count++;

}

}

return count;

}

int main() {

int size;

cout << "Enter size of array: ";

cin >> size;

int\* numbers = new int[size];

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

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

cin >> numbers[i];

}

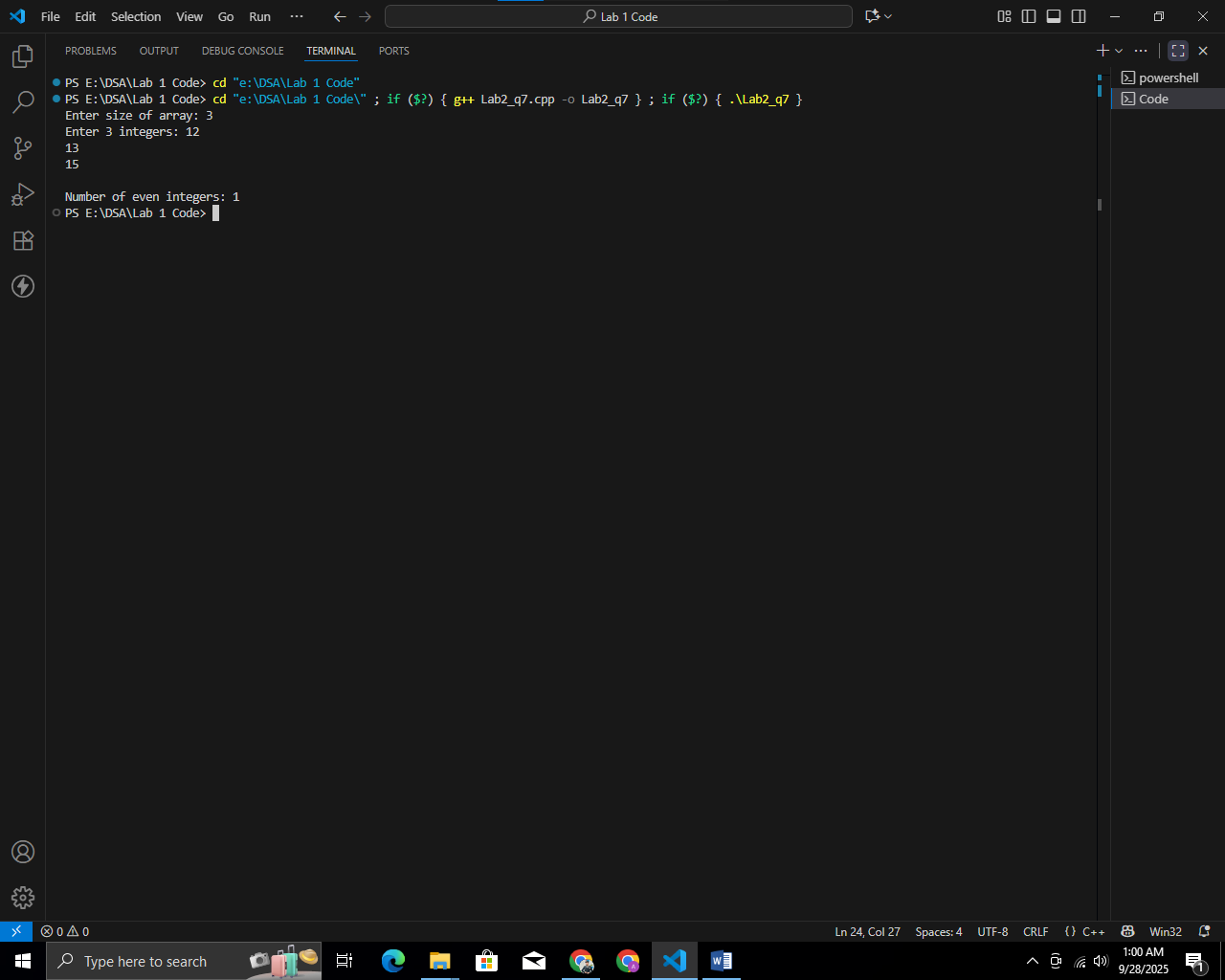
int evenCount = countEven(numbers, size);

cout << "\nNumber of even integers: " << evenCount << endl;

delete[] numbers;

return 0;

}



Q8. Write a function revString(char\*) which reverses the parameter. The function returns nothing. int main() { char s[10] = "somestring"; revString(s); // call the function return 0; } void revtString(char\* ptr) { // WRITE YOUR CODE HERE }

#include <stdio.h>

#include <string.h>

void revString(char \*ptr) {

int len = strlen(ptr);

int i, j;

char temp;

for (i = 0, j = len - 1; i < j; i++, j--) {

temp = ptr[i];

ptr[i] = ptr[j];

ptr[j] = temp;

}

}

int main() {

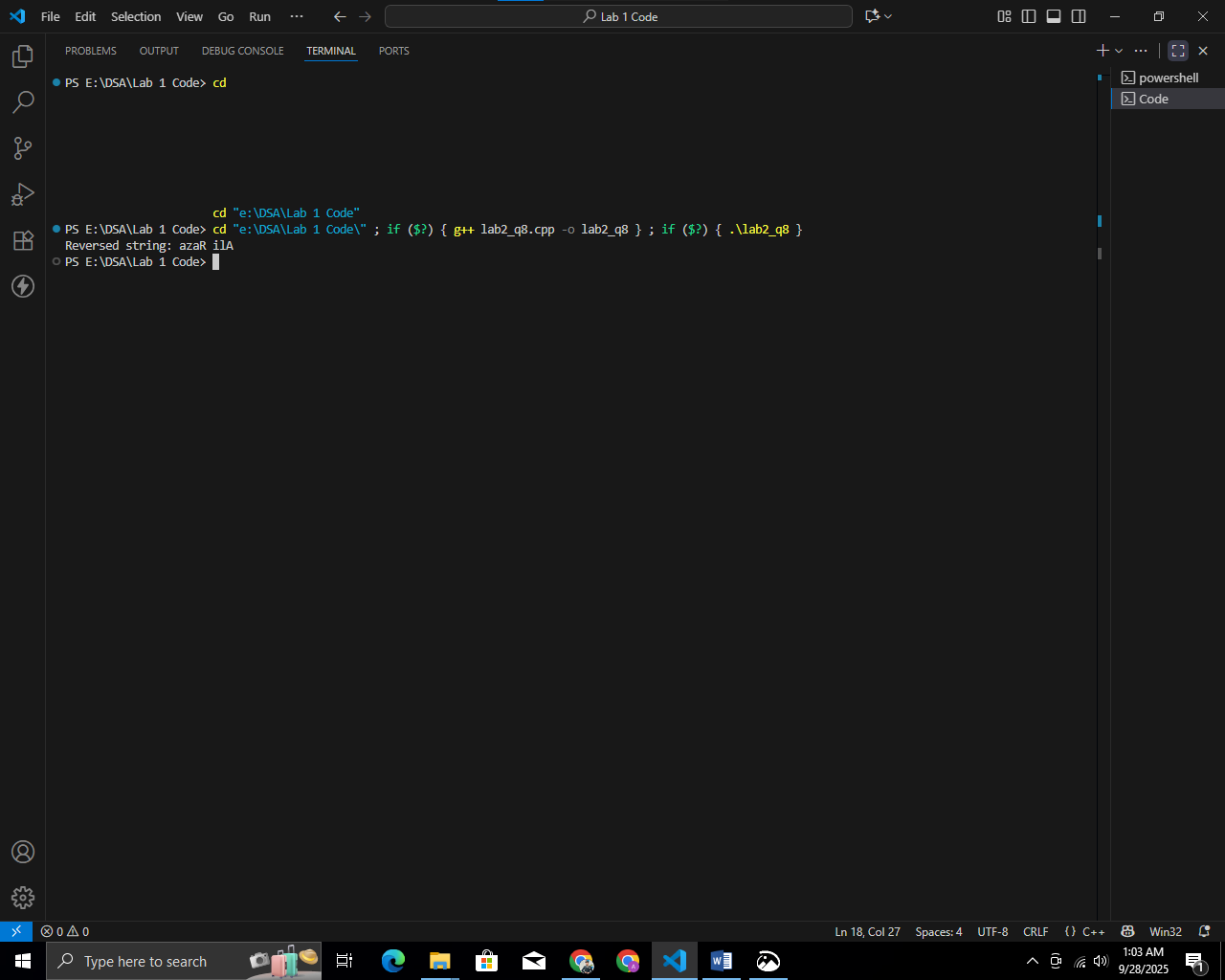
char s[20] = "Ali Raza";

revString(s);

printf("Reversed string: %s\n", s);

return 0;

}



Q9. Write a program that uses pointers to swap the values of two integers entered by the user.

#include <stdio.h>

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int x, y;

printf("Enter first integer: ");

scanf("%d", &x);

printf("Enter second integer: ");

scanf("%d", &y);

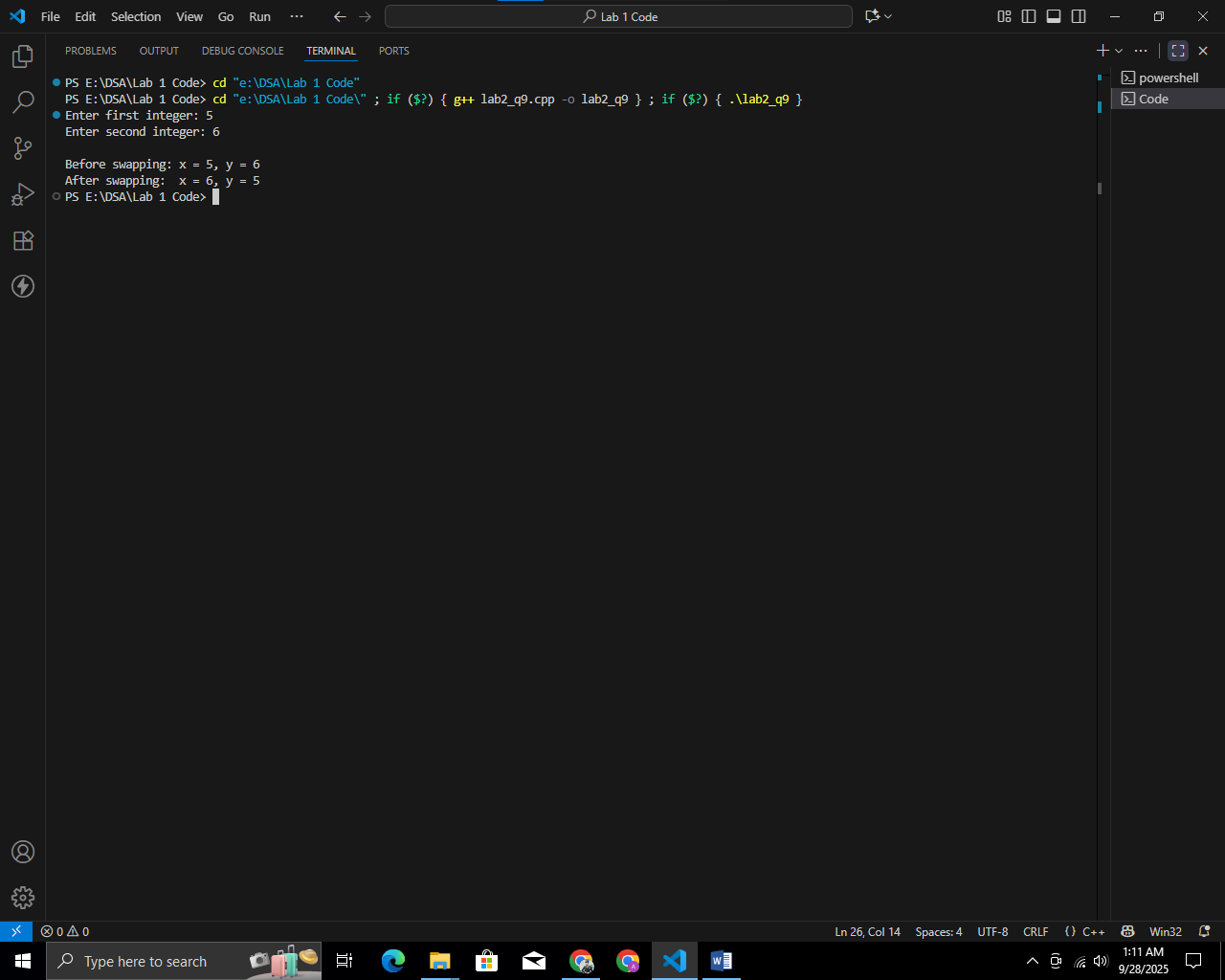
printf("\nBefore swapping: x = %d, y = %d\n", x, y);

swap(&x, &y);

printf("After swapping: x = %d, y = %d\n", x, y);

return 0;

}



Q10. Create a function that takes two integers as parameters and returns their sum using pointers.

#include <stdio.h>

int sum(int \*a, int \*b) {

return \*a + \*b;

}

int main() {

int x, y, result;

printf("Enter first integer: ");

scanf("%d", &x);

printf("Enter second integer: ");

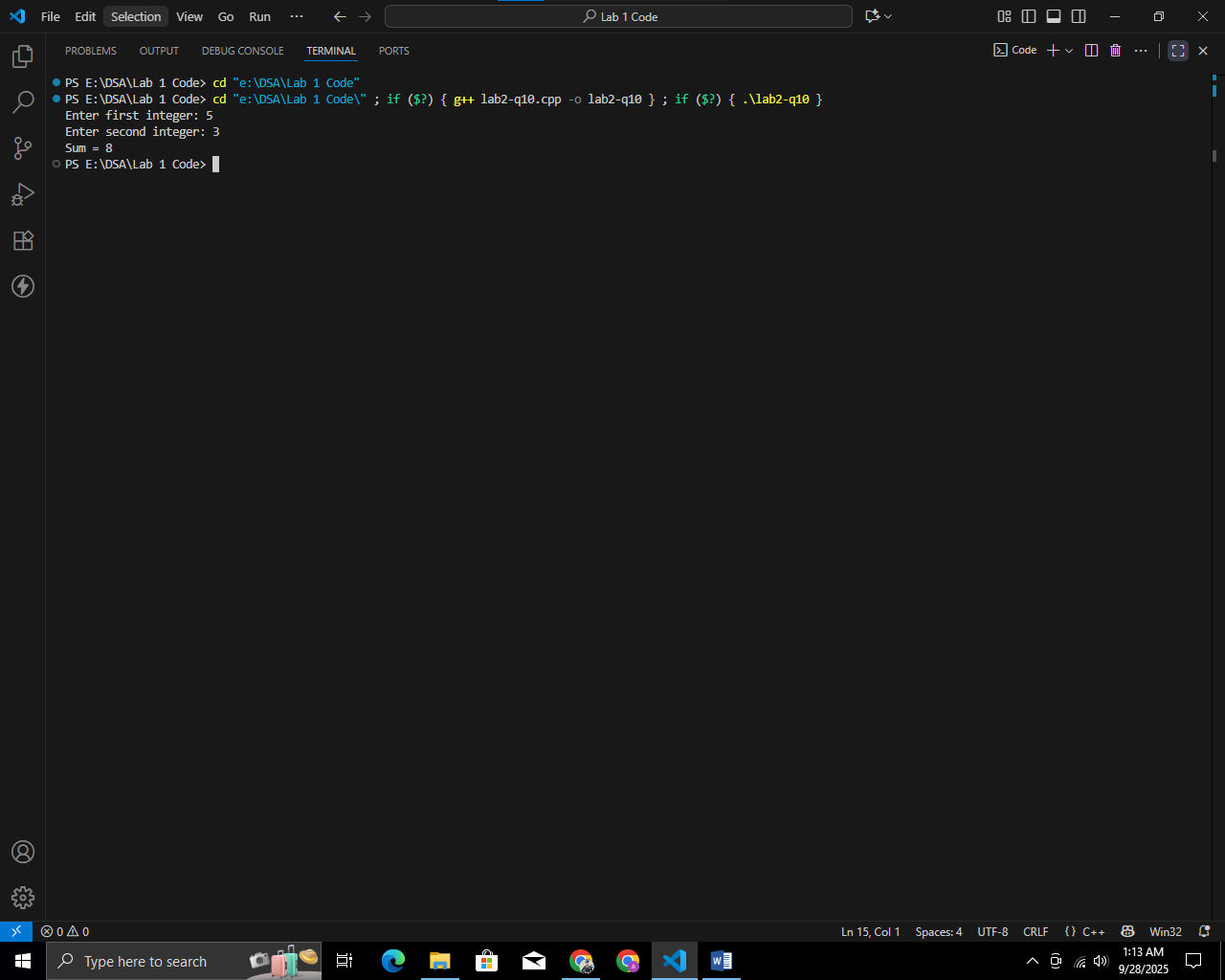
scanf("%d", &y);

result = sum(&x, &y);

printf("Sum = %d\n", result);

return 0;

}



Q11. Implement a program that uses an array of pointers to store and print the names of three students entered by the user

#include <stdio.h>

int main() {

static char name1[50], name2[50], name3[50];

char \*names[3] = {name1, name2, name3};

for (int i = 0; i < 3; i++) {

printf("Enter name of student %d: ", i + 1);

scanf("%49s", names[i]);

}

printf("\nYou entered:\n");

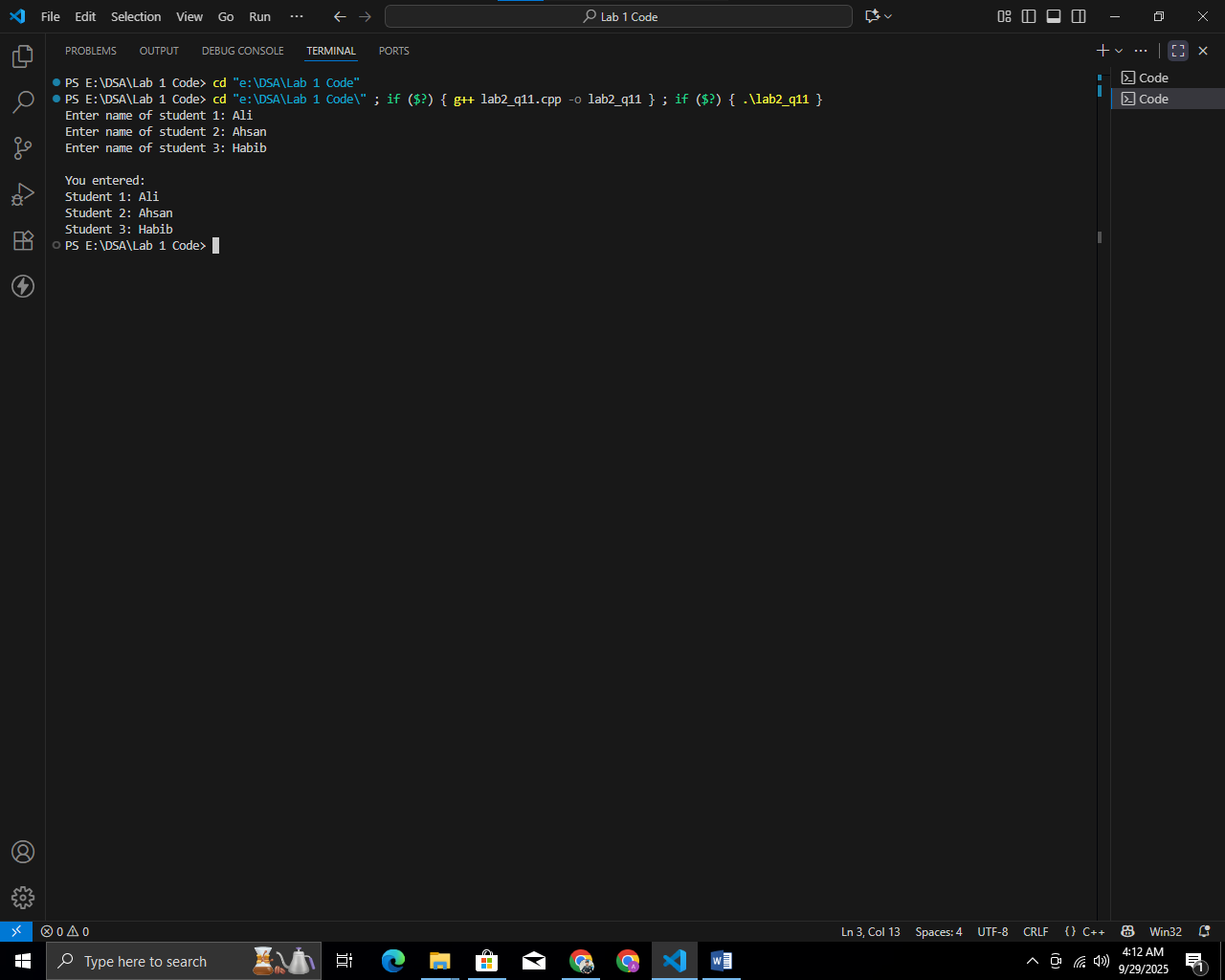
for (int i = 0; i < 3; i++) {

printf("Student %d: %s\n", i + 1, names[i]);

}

return 0;

}



Q12. Design a function that takes a pointer to a constant integer as a parameter and prints the value it points to

#include <stdio.h>

void printValue(const int \*ptr) {

printf("Value: %d\n", \*ptr);

}

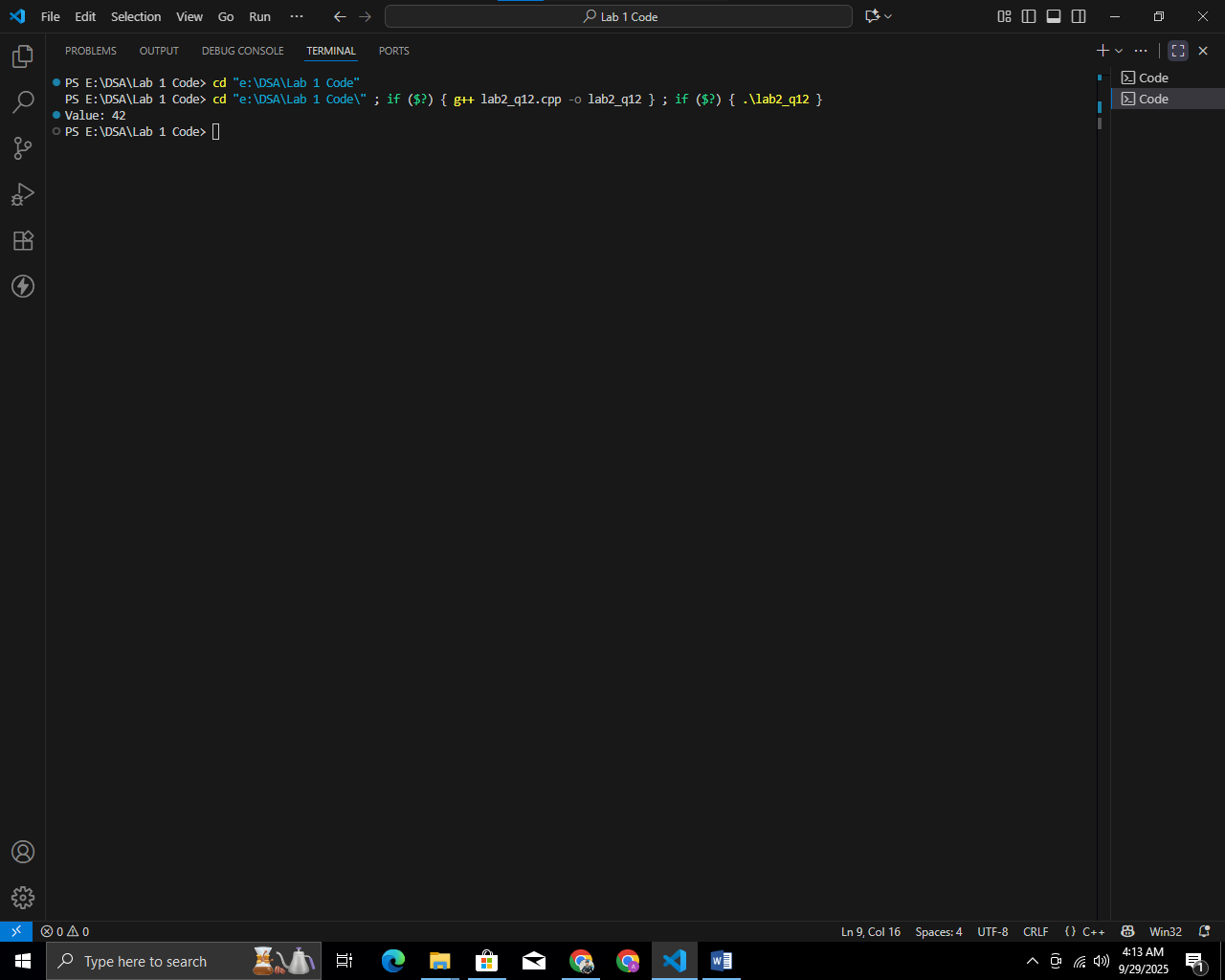
int main() {

int x = 42;

printValue(&x);

return 0;

}



Q13. Implement a set of overloaded functions that calculate the area of a square, rectangle, and circle using pointers

#include <iostream>

using namespace std;

double area(const double \*side) {

return (\*side) \* (\*side);

}

double area(const double \*length, const double \*width) {

return (\*length) \* (\*width);

}

double area(const double \*radius, bool circle) {

const double pi = 3.141592653589793;

return pi \* (\*radius) \* (\*radius);

}

int main() {

double s, l, w, r;

cout << "Enter side of square: ";

cin >> s;

cout << "Area of square = " << area(&s) << endl;

cout << "Enter length and width of rectangle: ";

cin >> l >> w;

cout << "Area of rectangle = " << area(&l, &w) << endl;

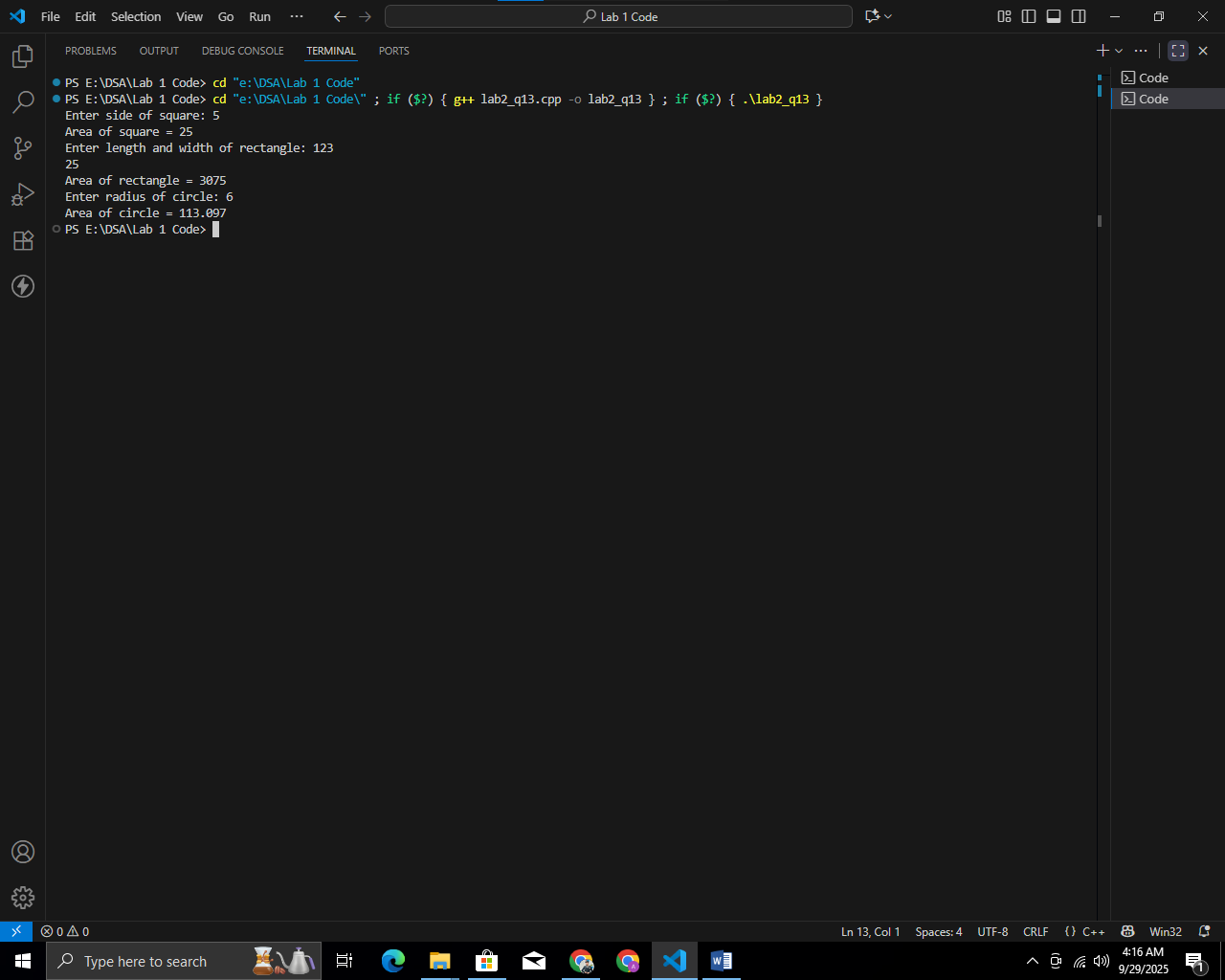
cout << "Enter radius of circle: ";

cin >> r;

cout << "Area of circle = " << area(&r, true) << endl;

return 0;

}



Q14. Develop a program that uses pointer arithmetic to find the length of a character array entered by the user.

#include <iostream>

using namespace std;

int characterArray(char \*p)

{

int i=0;

while (p[i]!='\0')

{

i++;

}

return i;

}

int main()

{

char arr[100];

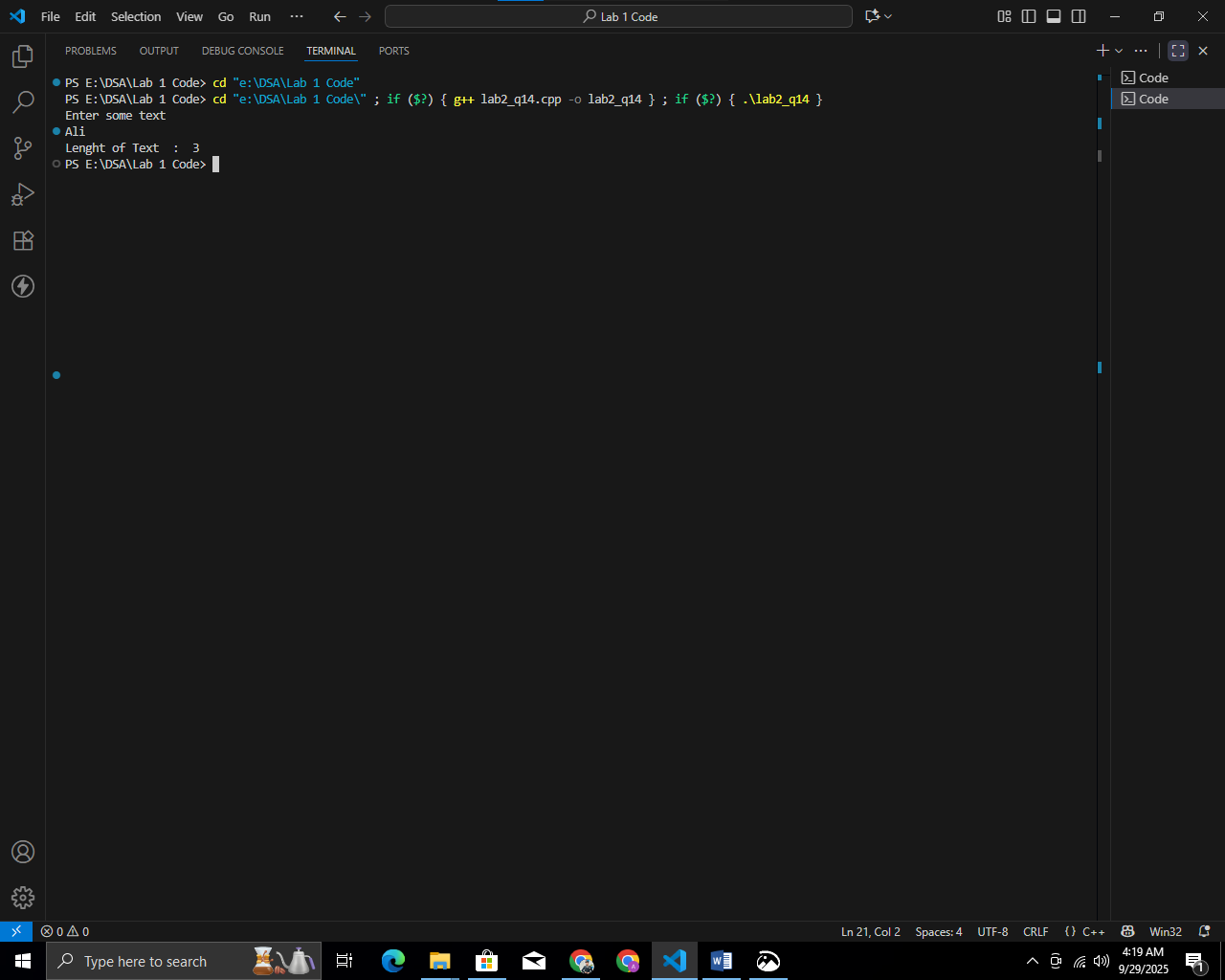
cout<<"Enter some text \n";

cin.getline(arr, 100);

cout<<"Lenght of Text : "<< characterArray(arr);

return 0;

}



Q15. Write a function that accepts an array of integers and its size as parameters. The function should find and print the minimum value using pointers.

#include <iostream>

#include <climits>

using namespace std;

int \* findMinOfArray(int arr[],int size)

{

int ans = INT\_MAX;

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

{

if(arr[i]<ans)

{

ans=arr[i];

}

}

int \*mini=new int(ans);

return mini;

}

int main()

{

int size=10;

int arr[10]={1,234,45,434,53534,35,6545,343,645,3245};

cout<<"Min of Array : "<<\* findMinOfArray(arr,size);

return 0;

}

