**Lab No.10**

1. Declare a structure named employee that stores the employee id, name, salary and department.

#include <stdio.h>

struct employee {

int id;

char name[50];

float salary;

char department[50];

};

int main() {

struct employee emp;

emp.id = 2500;

sprintf(emp.name, "sahil");

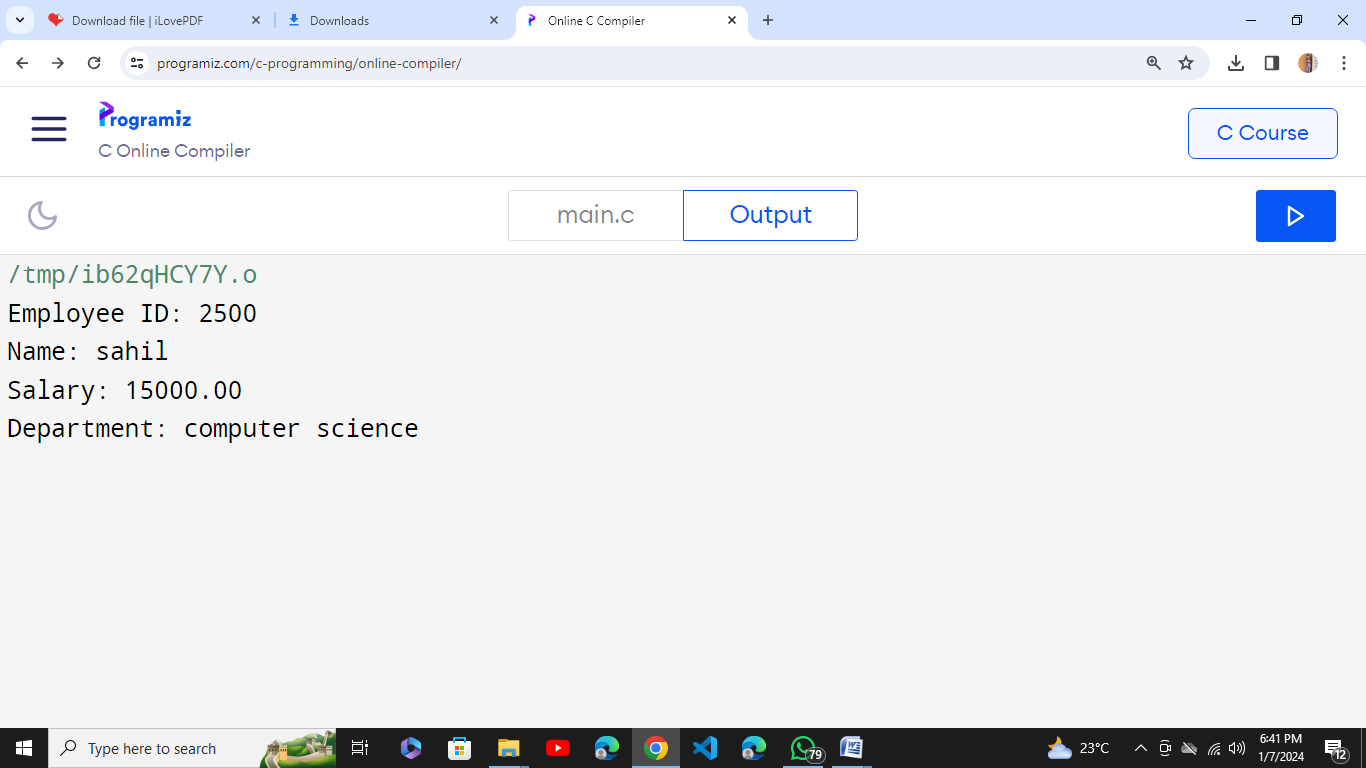
emp.salary = 15000.0;

sprintf(emp.department, "computer science");

printf("Employee ID: %d\n", emp.id);

printf("Name: %s\n", emp.name); **Out put:**

printf("Salary: %.2f\n", emp.salary);



printf("Department: %s\n", emp.department);

return 0;

}

2. Take data input from user after declaring a variable of employee type and show the data in proper format on output screen.

#include <stdio.h>

struct Employee {

char name[50];

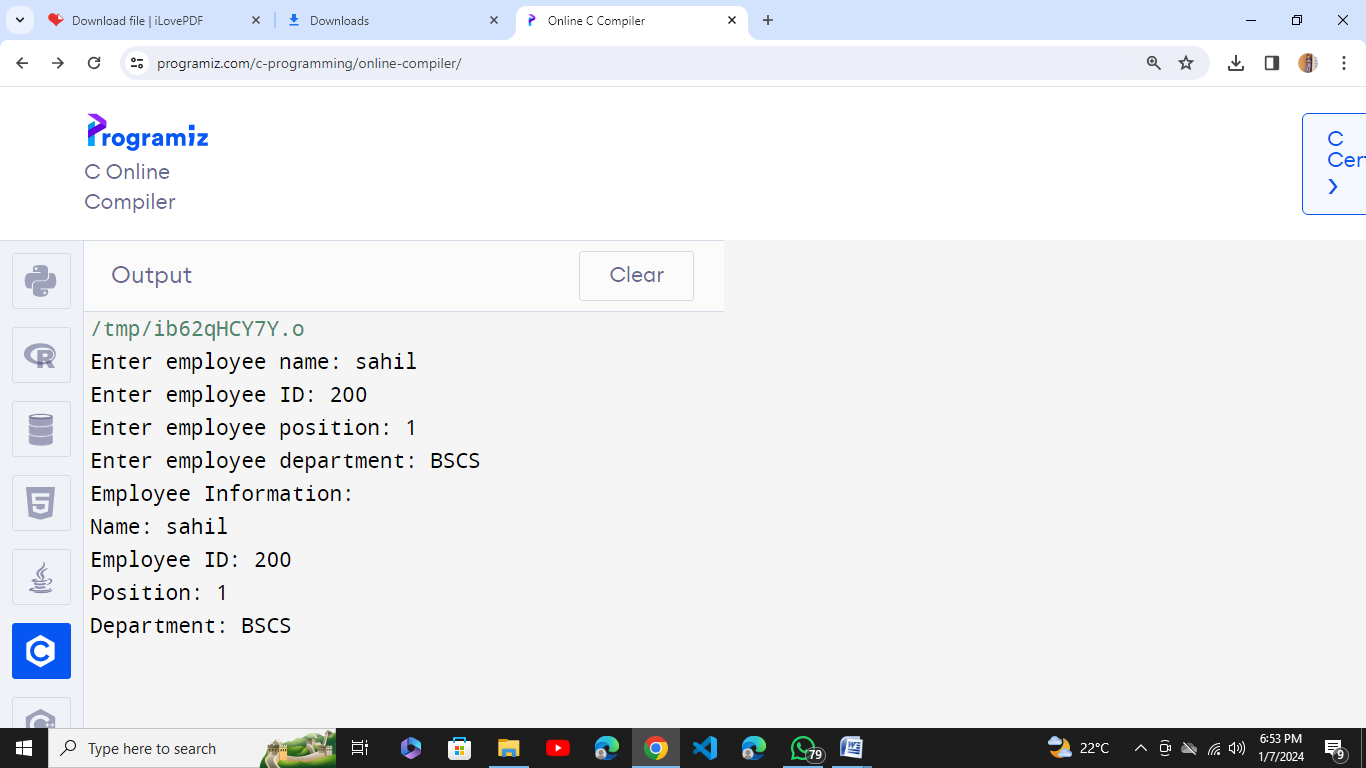
int employee\_id;

char position[50];

char department[50];

}; **Out put:**

int main() {



struct Employee emp;

printf("Enter employee name: ");

scanf("%s", emp.name);

printf("Enter employee ID: ");

scanf("%d", &emp.employee\_id);

printf("Enter employee position: ");

scanf("%s", emp.position);

printf("Enter employee department: ");

scanf("%s", emp.department);

printf("\nEmployee Information:\n");

printf("Name: %s\n", emp.name);

printf("Employee ID: %d\n", emp.employee\_id);

printf("Position: %s\n", emp.position);

printf("Department: %s\n", emp.department);

return 0;

}

3. A phone number, such as (212) 767-8900, can be thought of as having three parts: e.g., the area code (212), the exchange (767), and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers.

The interchange might look like this:

Enter area code: 415

Enter exchange: 555

Enter number: 1212

Then display like below:

My number is (212) 767-8900

Your number is (415) 555-1212

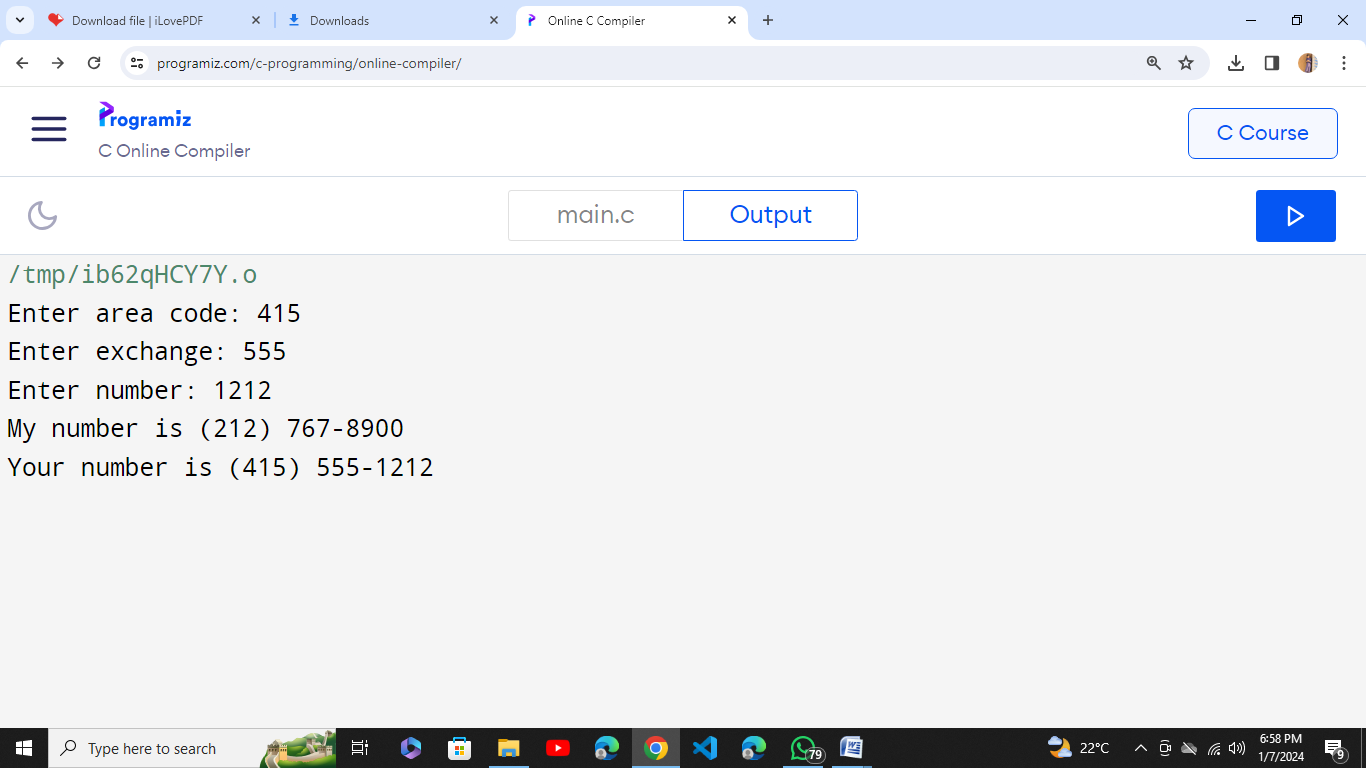
#include <stdio.h>

struct Phone {

int area\_code; **Out put:**

int exchange;

int number;



};

int main() {

struct Phone my\_number = {212, 767, 8900};

struct Phone your\_number;

printf("Enter area code: ");

scanf("%d", &your\_number.area\_code);

printf("Enter exchange: ");

scanf("%d", &your\_number.exchange);

printf("Enter number: ");

scanf("%d", &your\_number.number);

printf("My number is (%d) %d-%04d\n", my\_number.area\_code, my\_number.exchange, my\_number.number);

printf("Your number is (%d) %d-%04d\n", your\_number.area\_code, your\_number.exchange, your\_number.number);

return 0;

}

4. Define a structure to store the following student data: CGPA, courses (course name, GPA), address (consisting of street address, city, state, zip). Input 2 student records, compare and display which student have highest GPA in which course also Display which student has the highest CGPA . HINT: define another structure to hold the courses and address.

#include <stdio.h>

#include <string.h>

#define MAX\_COURSES 5

#define MAX\_NAME\_LENGTH 50

#define MAX\_ADDRESS\_LENGTH 100

struct Course {

char course\_name[MAX\_NAME\_LENGTH];

float GPA;

};

struct Address {

char street[MAX\_ADDRESS\_LENGTH];

char city[MAX\_NAME\_LENGTH];

char state[MAX\_NAME\_LENGTH];

char zip[MAX\_NAME\_LENGTH];

};

struct Student {

float CGPA;

struct Course courses[MAX\_COURSES];

struct Address address;

};

int main() {

struct Student student1, student2;

printf("Enter CGPA for student 1: ");

scanf("%f", &student1.CGPA);

printf("Enter student 1's address:\n");

printf("Street: ");

scanf("%s", student1.address.street);

printf("City: ");

scanf("%s", student1.address.city);

printf("State: ");

scanf("%s", student1.address.state);

printf("ZIP: ");

scanf("%s", student1.address.zip);

printf("Enter student 1's course details:\n");

for (int i = 0; i < MAX\_COURSES; ++i) {

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

scanf("%s", student1.courses[i].course\_name);

printf("Enter GPA for course %d: ", i + 1);

scanf("%f", &student1.courses[i].GPA);

}

printf("\nEnter CGPA for student 2: ");

scanf("%f", &student2.CGPA);

printf("Enter student 2's address:\n");

printf("Street: ");

scanf("%s", student2.address.street);

printf("City: ");

scanf("%s", student2.address.city);

printf("State: ");

scanf("%s", student2.address.state);

printf("ZIP: ");

scanf("%s", student2.address.zip);

printf("Enter student 2's course details:\n");

for (int i = 0; i < MAX\_COURSES; ++i) {

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

scanf("%s", student2.courses[i].course\_name);

printf("Enter GPA for course %d: ", i + 1);

scanf("%f", &student2.courses[i].GPA);

}

for (int i = 0; i < MAX\_COURSES; ++i) {

if (student1.courses[i].GPA > student2.courses[i].GPA) {

printf("\nStudent 1 has the highest GPA (%.2f) in course %s\n", student1.courses[i].GPA, student1.courses[i].course\_name);

} else if (student1.courses[i].GPA < student2.courses[i].GPA) {

printf("\nStudent 2 has the highest GPA (%.2f) in course %s\n", student2.courses[i].GPA, student2.courses[i].course\_name);

} else {

printf("\nBoth students have the same GPA (%.2f) in course %s\n", student1.courses[i].GPA, student1.courses[i].course\_name);

}

}

if (student1.CGPA > student2.CGPA) {

printf("\nStudent 1 has the highest CGPA: %.2f\n", student1.CGPA);

} else if (student1.CGPA < student2.CGPA) {

printf("\nStudent 2 has the highest CGPA: %.2f\n", student2.CGPA);

} else {

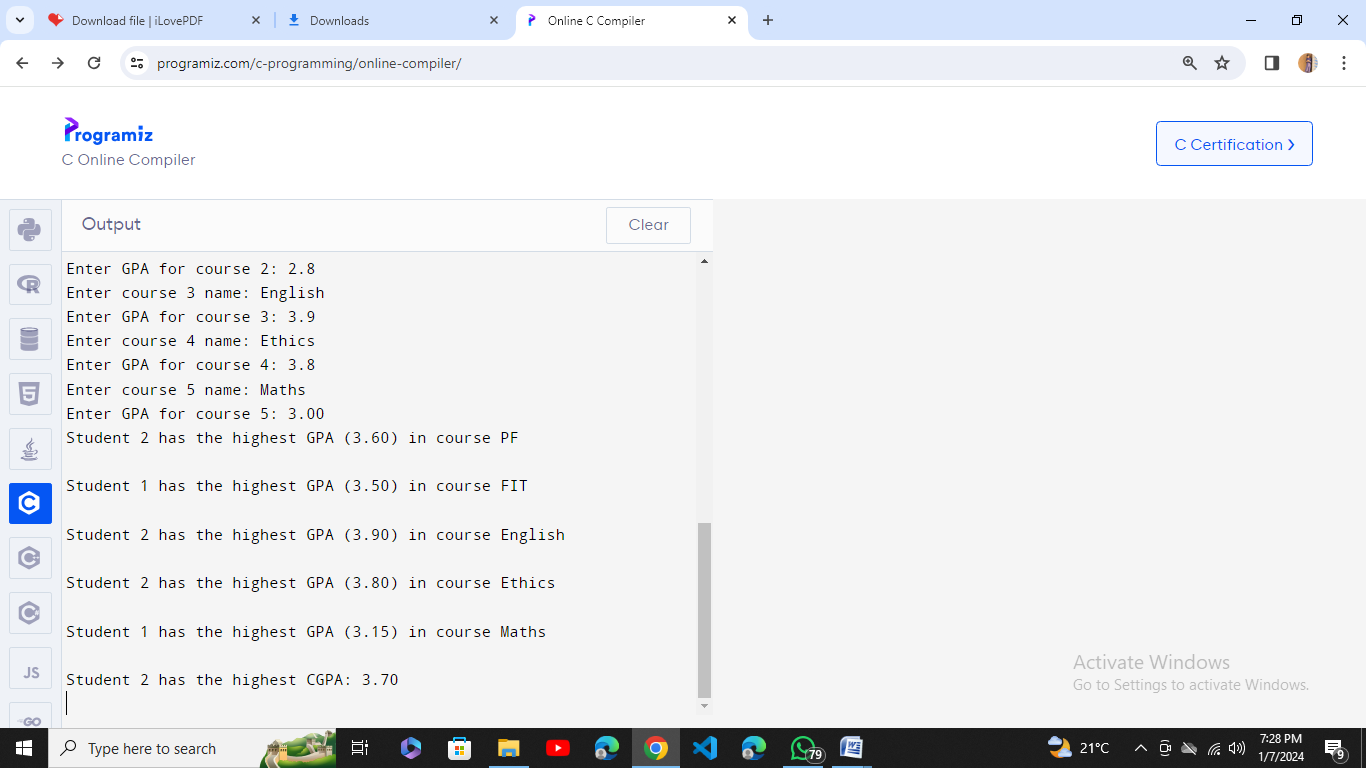
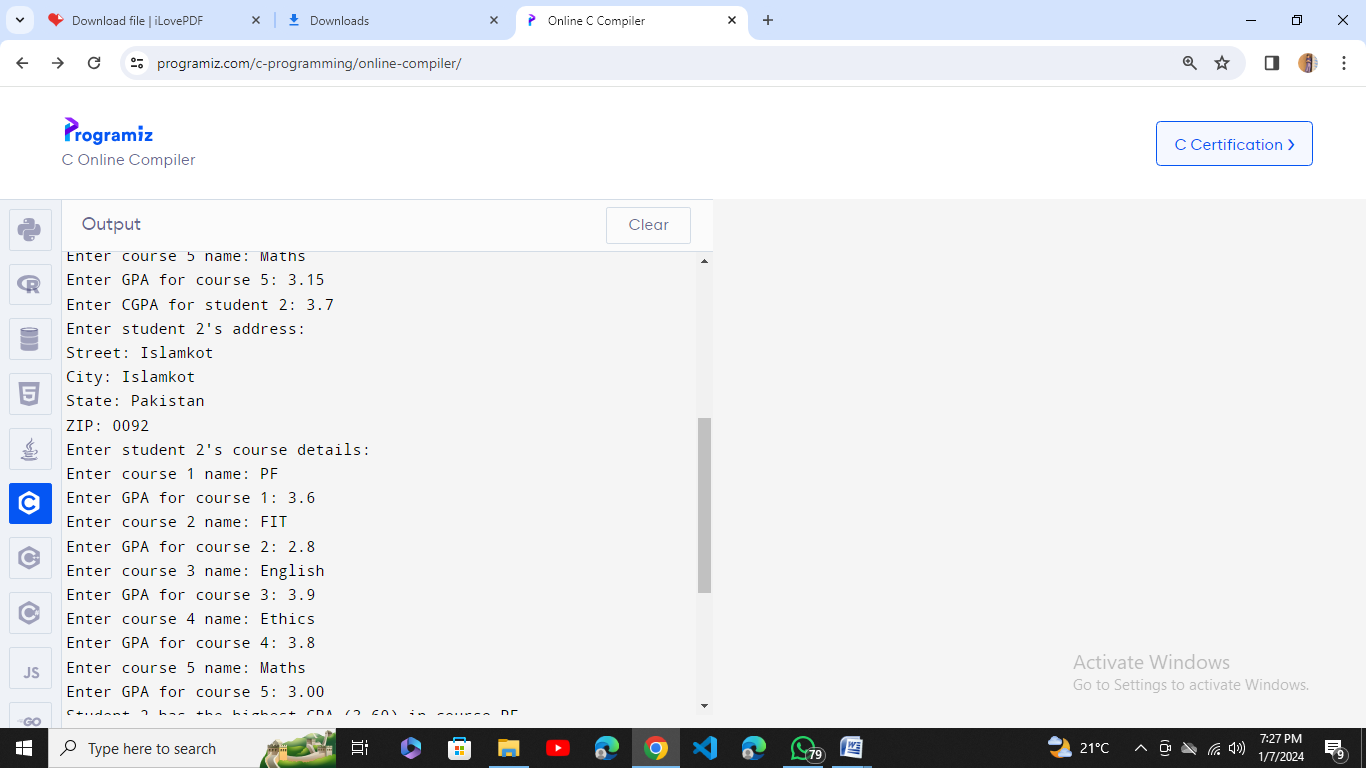
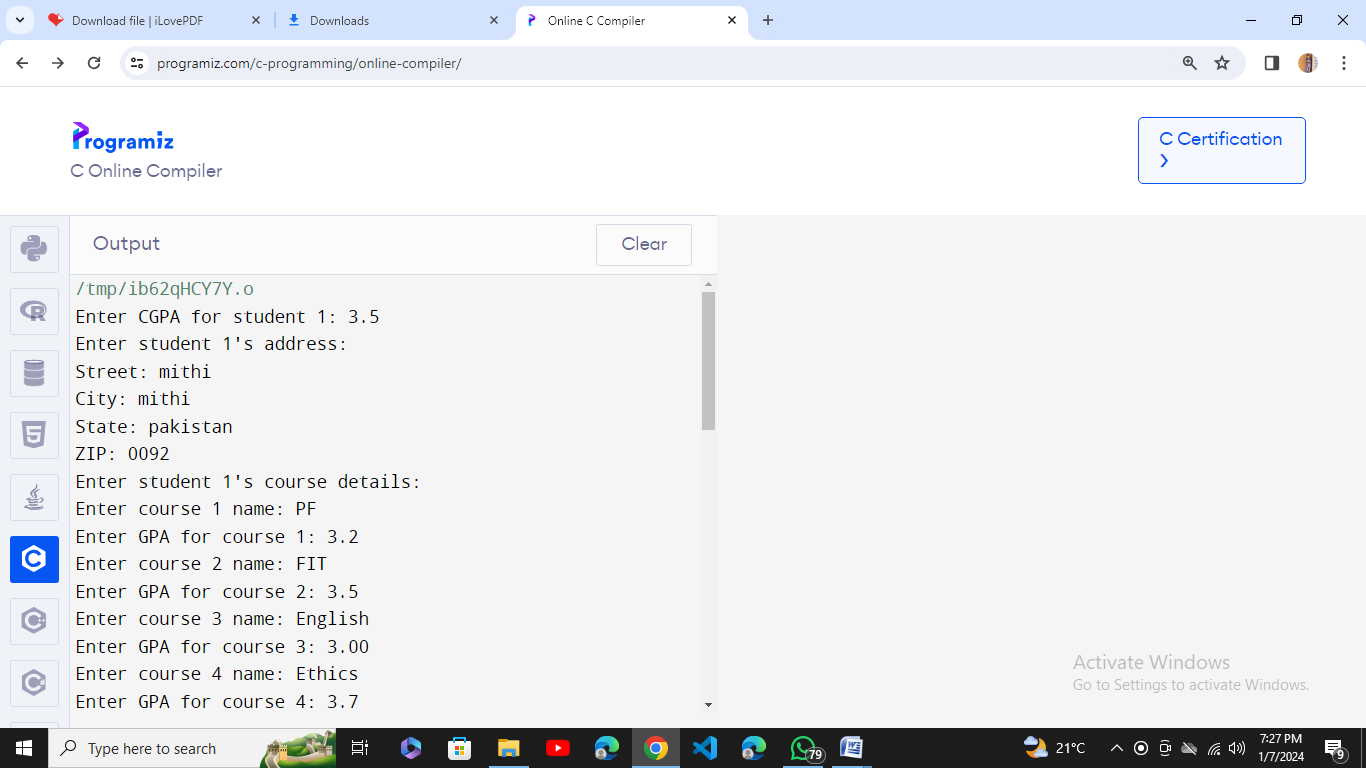
printf("\nBoth students have the same CGPA: %.2f\n", student1.CGPA);

}

return 0;

}

**Out put:**



5. Write a C program that uses functions to perform the following operations:

i) Reading a complex number

ii) Writing a complex number

iii) Addition of two complex numbers

iv) Multiplication of two complex numbers

(Note: represent complex numbers using a structure.)

#include <stdio.h>

struct Complex {

float real;

float imag;

};

struct Complex readComplex() {

struct Complex num;

printf("Enter real part: ");

scanf("%f", &num.real);

printf("Enter imaginary part: ");

scanf("%f", &num.imag);

return num;

}

void writeComplex(struct Complex num) {

if (num.imag >= 0) {

printf("%.2f + %.2fi\n", num.real, num.imag);

} else {

printf("%.2f - %.2fi\n", num.real, -num.imag);

}

}

struct Complex addComplex(struct Complex num1, struct Complex num2) {

struct Complex sum;

sum.real = num1.real + num2.real;

sum.imag = num1.imag + num2.imag;

return sum;

}

struct Complex multiplyComplex(struct Complex num1, struct Complex num2) {

struct Complex product;

product.real = (num1.real \* num2.real) - (num1.imag \* num2.imag);

product.imag = (num1.real \* num2.imag) + (num1.imag \* num2.real);

return product;

}

int main() {

struct Complex complex1, complex2, result;

printf("Enter first complex number:\n");

complex1 = readComplex();

printf("\nEnter second complex number:\n");

complex2 = readComplex();

printf("\nThe first complex number is: ");

writeComplex(complex1);

printf("The second complex number is: ");

writeComplex(complex2);

result = addComplex(complex1, complex2);

printf("\nSum of the complex numbers: ");

writeComplex(result);

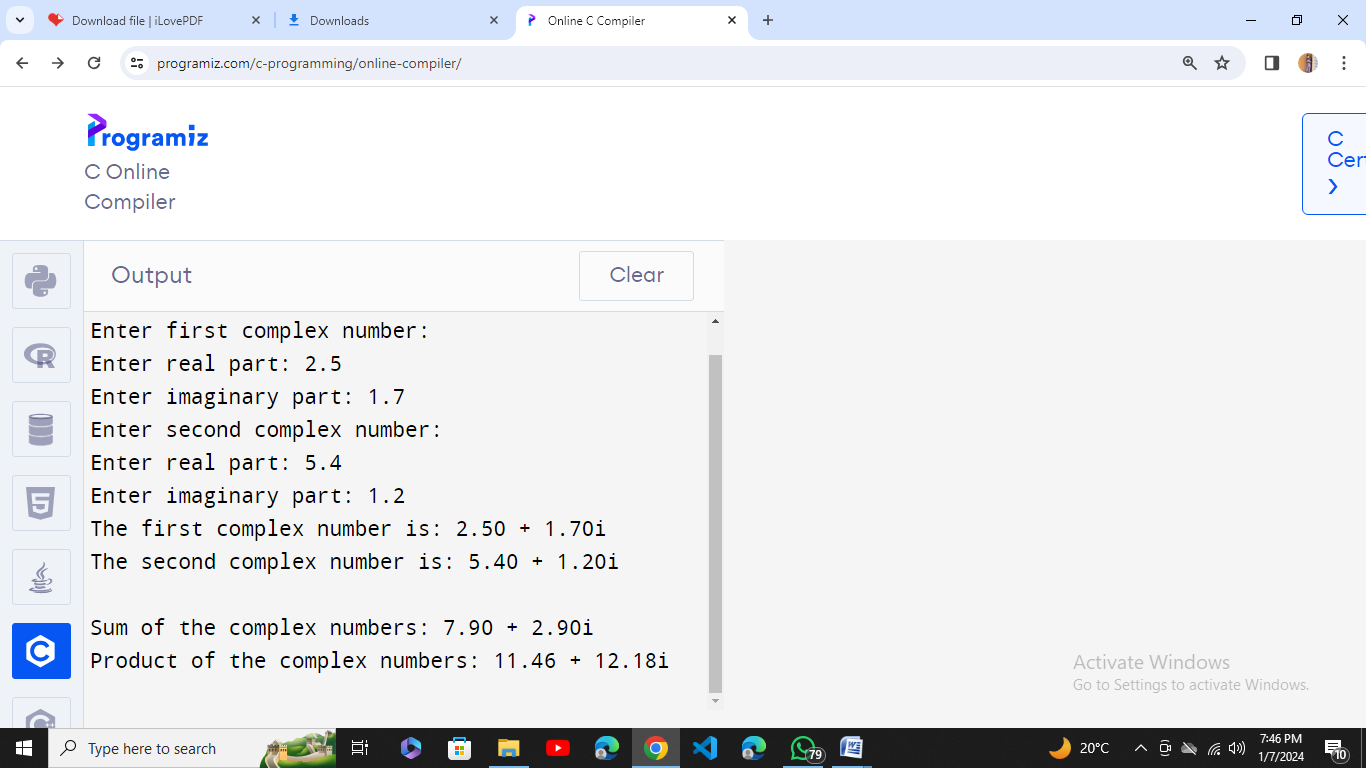
result = multiplyComplex(complex1, complex2);

printf("Product of the complex numbers: ");

writeComplex(result);

return 0;

}



**Out put:**