



Chandpur Science and Technology University

Dept. Of Computer Science & Engineering

CSE 1101

Assignment on #Structure

Submitted To

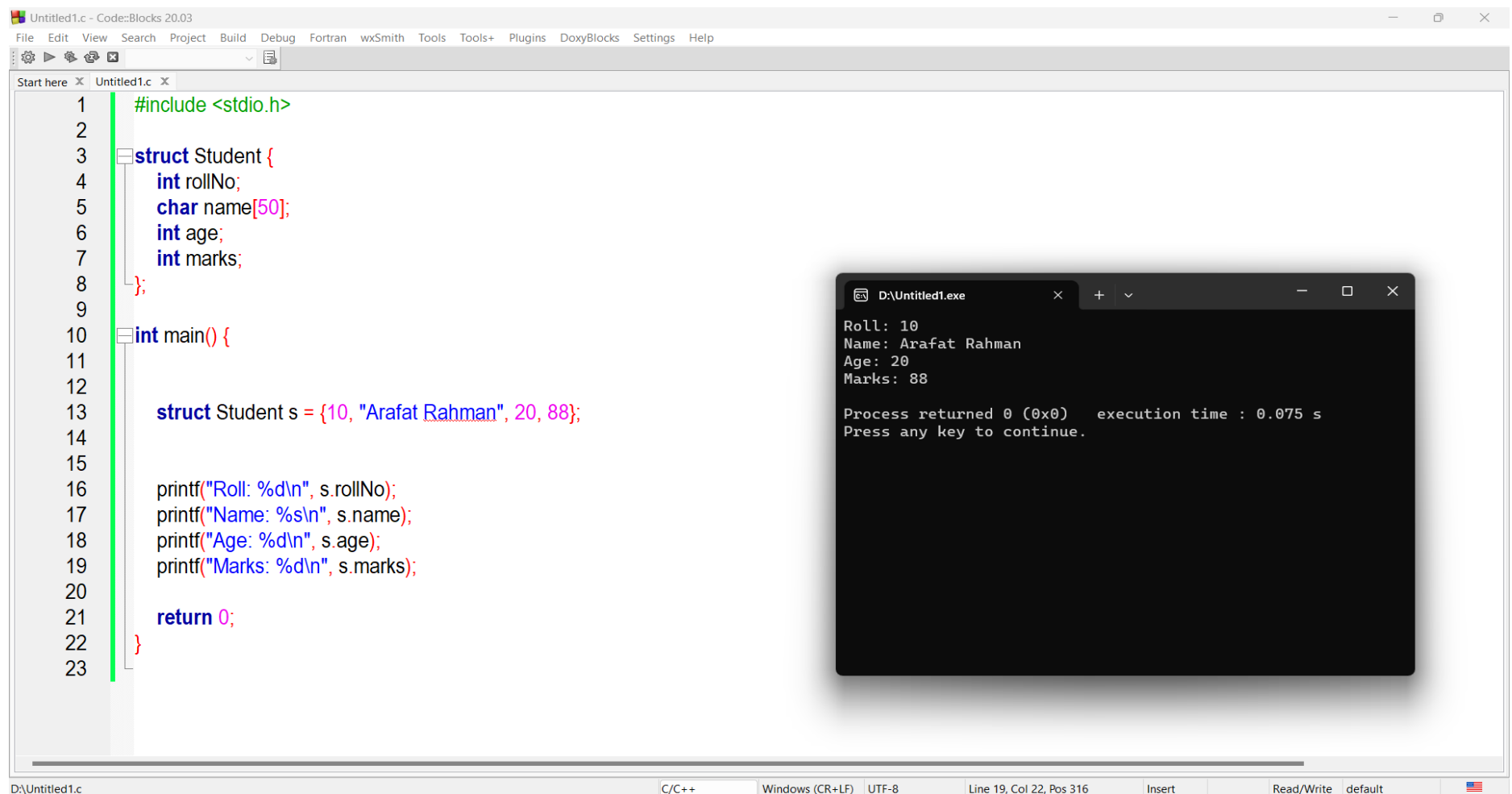
Mostafiz Ahammed
Lecturer, Dept. of CSE

Submitted By

Name- Iftekhar Hossain
ID - B220101024
Date of submission – 07/02/2024

Problem no – 1

Write a program to store(initialize) and print the roll no., name, age and marks of a student using structures.



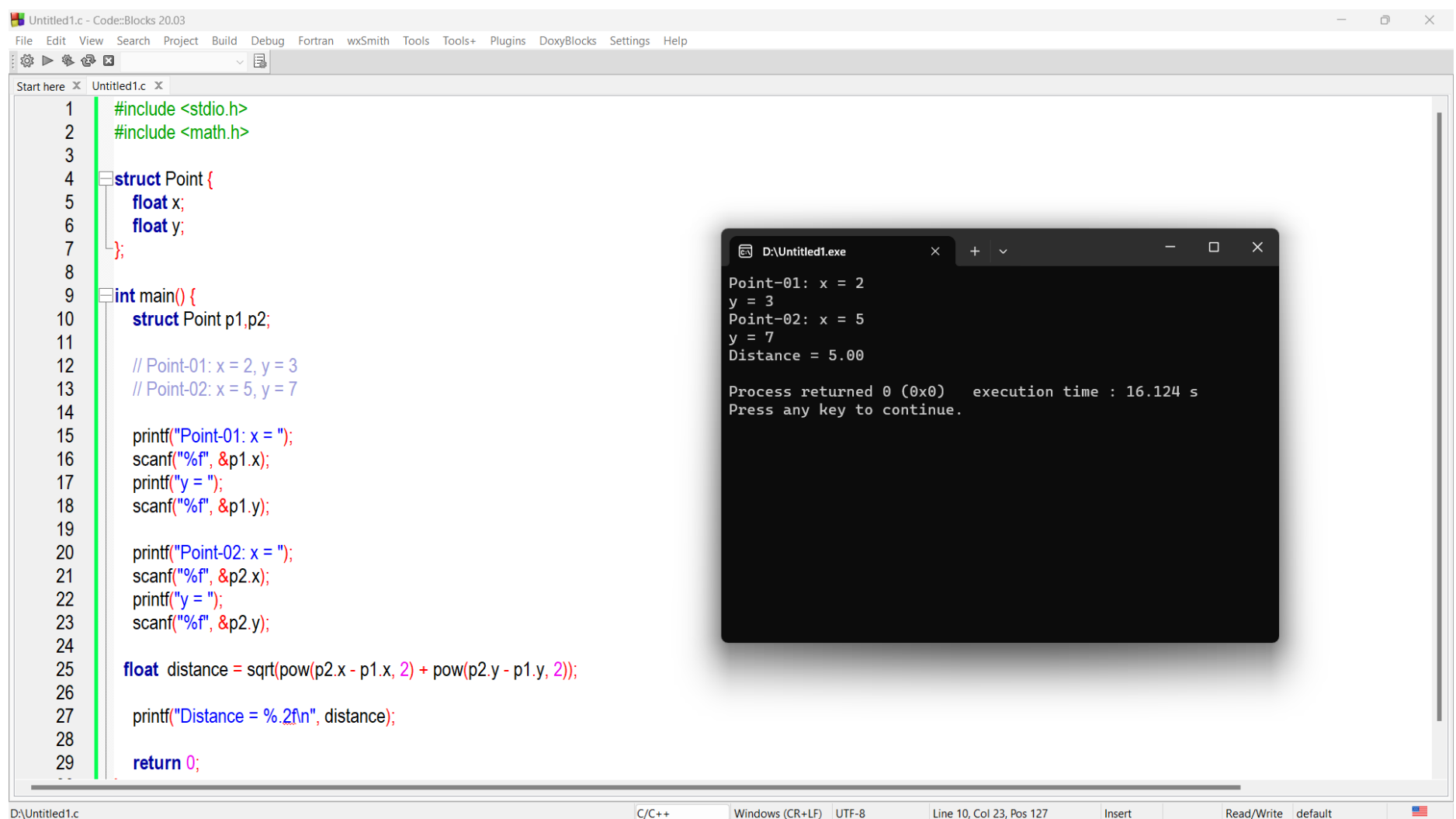
```
1 #include <stdio.h>
2
3 struct Student {
4     int rollNo;
5     char name[50];
6     int age;
7     int marks;
8 };
9
10 int main() {
11
12     struct Student s = {10, "Arafat Rahman", 20, 88};
13
14     printf("Roll: %d\n", s.rollNo);
15     printf("Name: %s\n", s.name);
16     printf("Age: %d\n", s.age);
17     printf("Marks: %d\n", s.marks);
18
19     return 0;
20 }
```

Roll: 10
Name: Arafat Rahman
Age: 20
Marks: 88

Process returned 0 (0x0) execution time : 0.075 s
Press any key to continue.

Problem no – 2

Write a program to find distance between two points using structure. The values of the points are to be taken from the user.



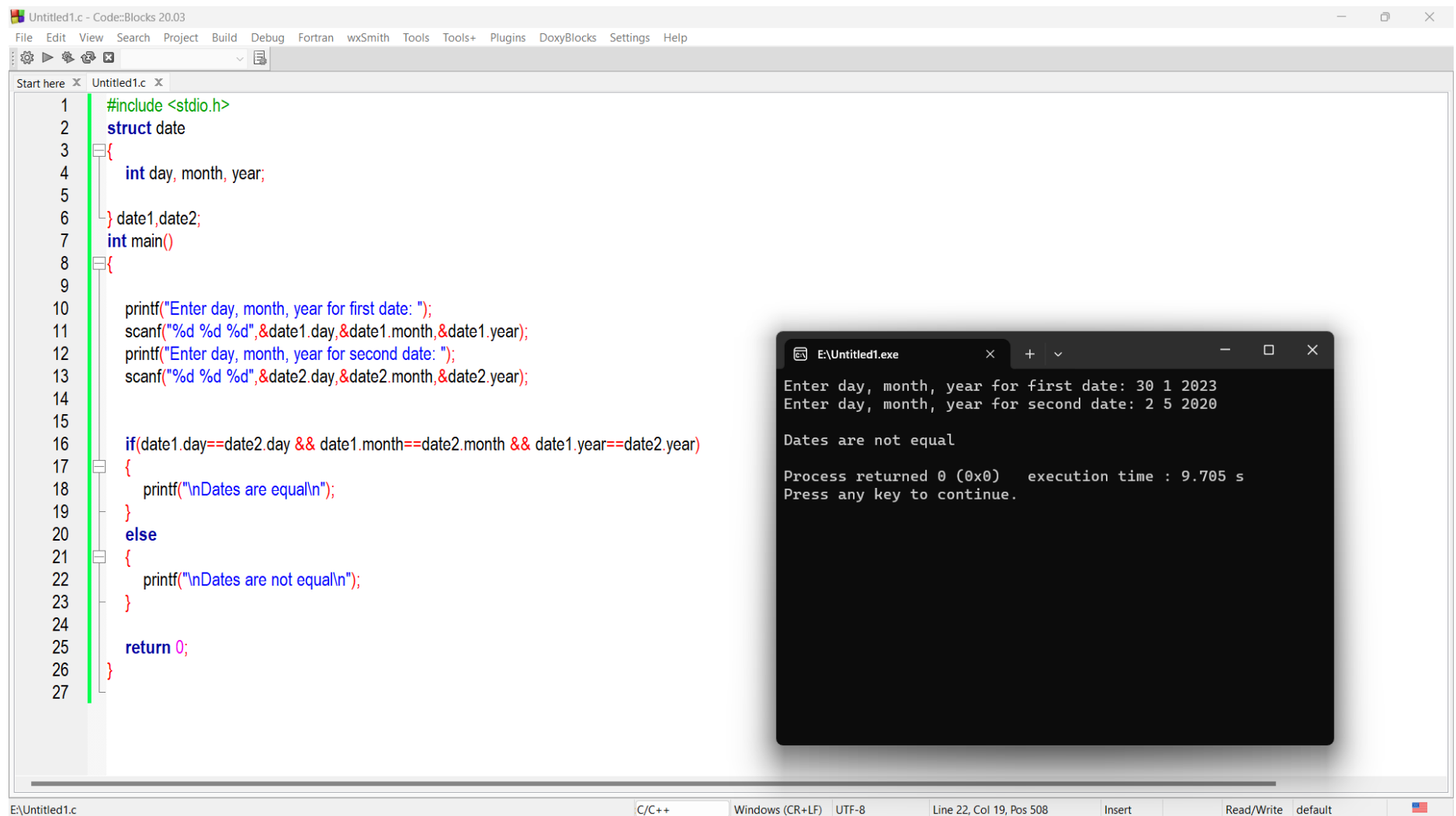
```
1 #include <stdio.h>
2 #include <math.h>
3
4 struct Point {
5     float x;
6     float y;
7 };
8
9 int main() {
10     struct Point p1,p2;
11
12     // Point-01: x = 2, y = 3
13     // Point-02: x = 5, y = 7
14
15     printf("Point-01: x = ");
16     scanf("%f", &p1.x);
17     printf("y = ");
18     scanf("%f", &p1.y);
19
20     printf("Point-02: x = ");
21     scanf("%f", &p2.x);
22     printf("y = ");
23     scanf("%f", &p2.y);
24
25     float distance = sqrt(pow(p2.x - p1.x, 2) + pow(p2.y - p1.y, 2));
26
27     printf("Distance = %.2f\n", distance);
28
29     return 0;
30 }
```

Point-01: x = 2
y = 3
Point-02: x = 5
y = 7
Distance = 5.00

Process returned 0 (0x0) execution time : 16.124 s
Press any key to continue.

Problem no – 3

Write a program to compare two dates entered by user. Make a structure named Date to store the elements day, month and year to store the dates. If the dates are equal, display "Dates are equal" otherwise display "Dates are not equal".



The screenshot shows the Code::Blocks IDE with a C program to compare two dates. The program defines a struct 'date' with fields 'day', 'month', and 'year'. It prompts the user to enter two dates and compares them. If the dates are equal, it prints "Dates are equal"; otherwise, it prints "Dates are not equal". The execution output shows the user entering the first date as 30 1 2023 and the second date as 2 5 2020, resulting in the output "Dates are not equal".

```
1 #include <stdio.h>
2 struct date
3 {
4     int day, month, year;
5 }
6 date1, date2;
7 int main()
8 {
9
10     printf("Enter day, month, year for first date: ");
11     scanf("%d %d %d", &date1.day, &date1.month, &date1.year);
12     printf("Enter day, month, year for second date: ");
13     scanf("%d %d %d", &date2.day, &date2.month, &date2.year);
14
15     if(date1.day==date2.day && date1.month==date2.month && date1.year==date2.year)
16     {
17         printf("\nDates are equal\n");
18     }
19     else
20     {
21         printf("\nDates are not equal\n");
22     }
23
24     return 0;
25 }
```

Execution Output:

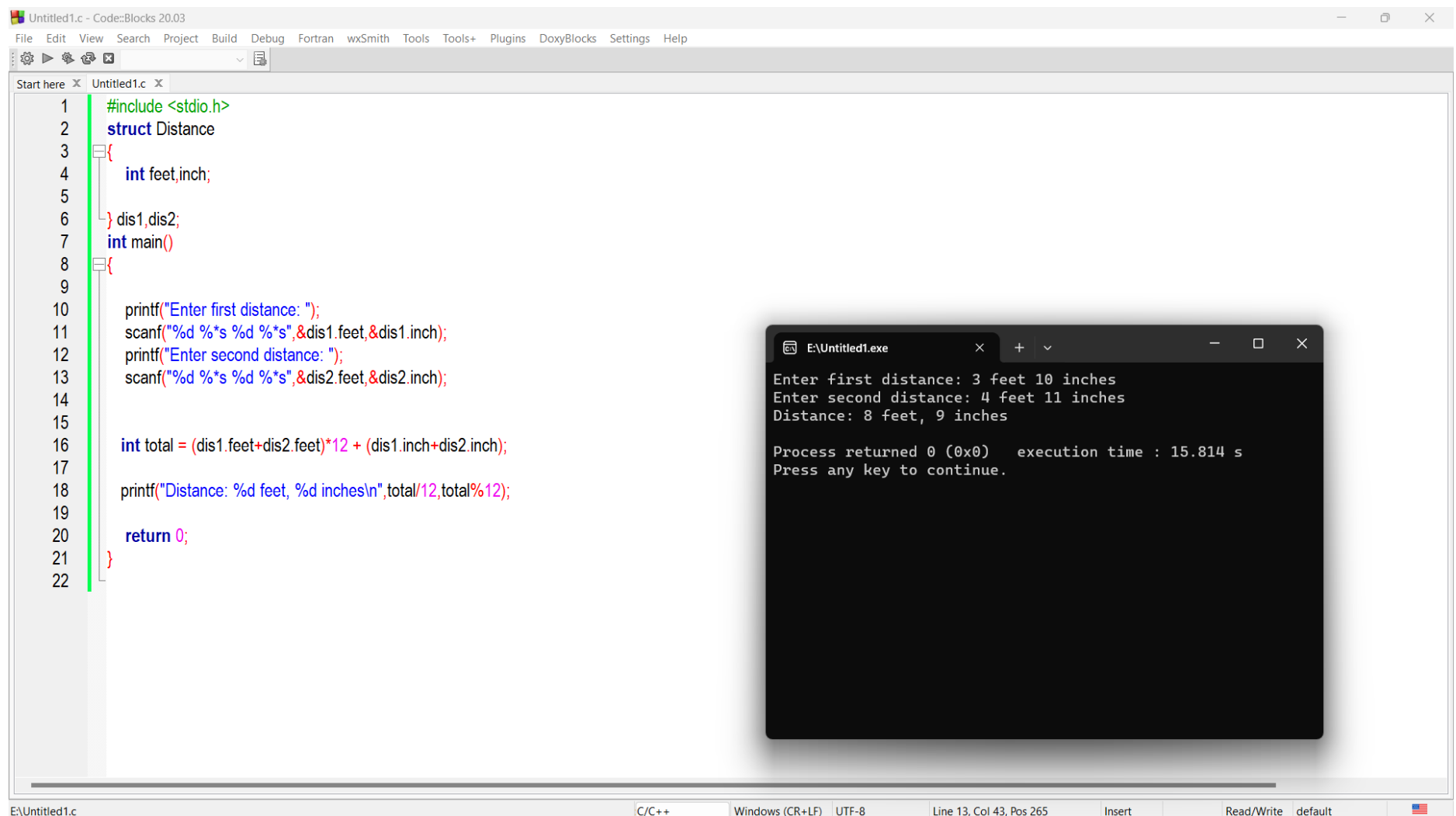
```
E:\Untitled1.exe
Enter day, month, year for first date: 30 1 2023
Enter day, month, year for second date: 2 5 2020

Dates are not equal

Process returned 0 (0x0)   execution time : 9.705 s
Press any key to continue.
```

Problem no – 4

Write a program to add two distances in inch-feet using structure. The values of the distances are to be taken from the user.



The screenshot shows the Code::Blocks IDE with a C program to add two distances in inch-feet. The program defines a struct 'Distance' with fields 'feet' and 'inch'. It prompts the user to enter two distances and calculates the total distance. The execution output shows the user entering the first distance as 3 feet 10 inches and the second distance as 4 feet 11 inches, resulting in the output "Distance: 8 feet, 9 inches".

```
1 #include <stdio.h>
2 struct Distance
3 {
4     int feet, inch;
5 }
6 dis1, dis2;
7 int main()
8 {
9
10     printf("Enter first distance: ");
11     scanf("%d %s %d %s", &dis1.feet, &dis1.inch);
12     printf("Enter second distance: ");
13     scanf("%d %s %d %s", &dis2.feet, &dis2.inch);
14
15     int total = (dis1.feet+dis2.feet)*12 + (dis1.inch+dis2.inch);
16
17     printf("Distance: %d feet, %d inches\n", total/12, total%12);
18
19     return 0;
20 }
```

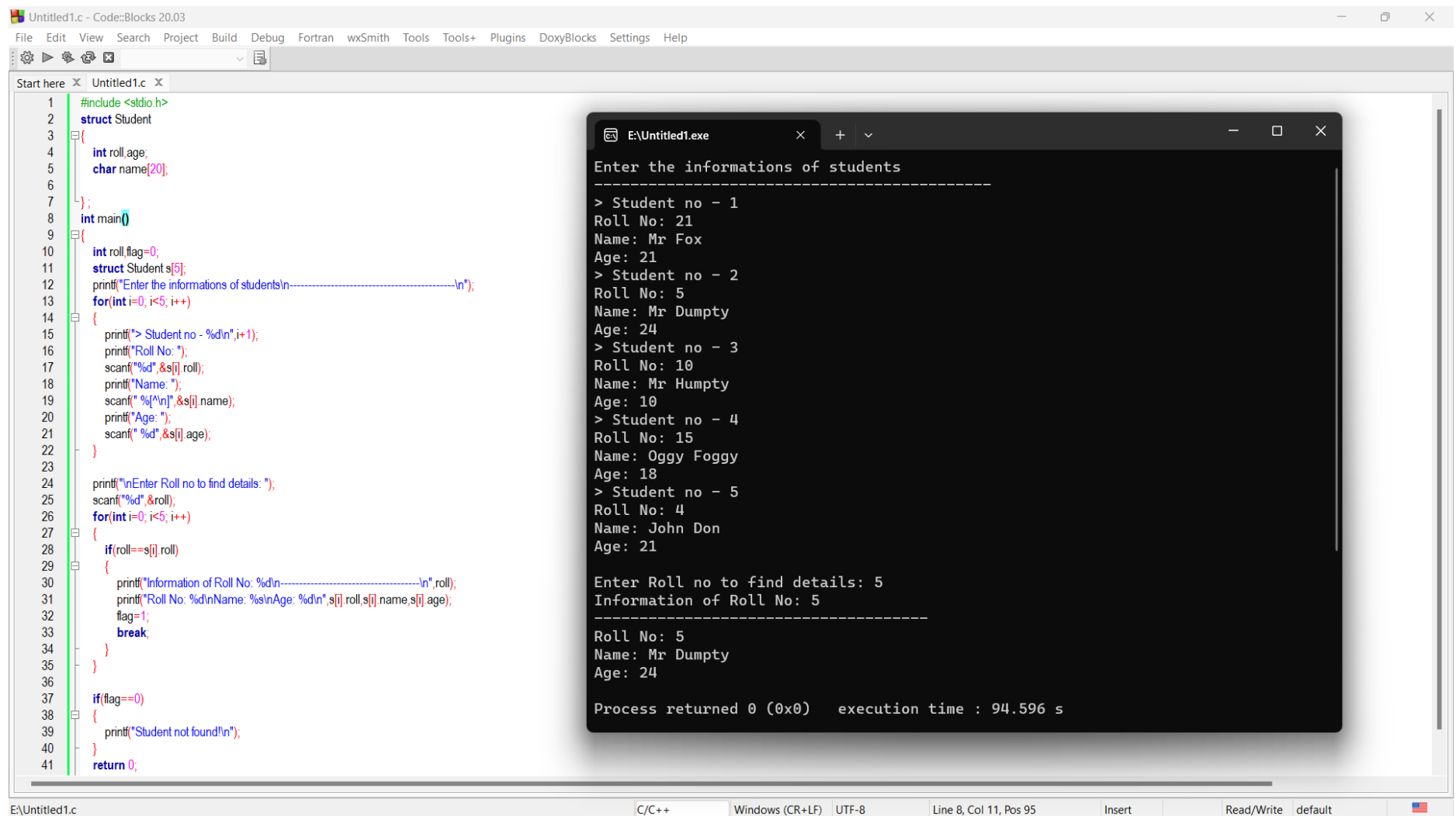
Execution Output:

```
E:\Untitled1.exe
Enter first distance: 3 feet 10 inches
Enter second distance: 4 feet 11 inches
Distance: 8 feet, 9 inches

Process returned 0 (0x0)   execution time : 15.814 s
Press any key to continue.
```

Problem no – 5

Write a program to store the roll no., name and age of 5 students and then print the details of the student with roll no. 2 or a specific roll number.



The screenshot shows the Code::Blocks IDE with a C program for Problem 5. The code defines a `Student` structure with `roll`, `age`, and `name` fields. It uses a loop to input data for 5 students and another loop to find and display details for a specific roll number (5 in the example).

```
1 #include <stdio.h>
2 struct Student
3 {
4     int roll, age;
5     char name[20];
6 }
7
8 int main()
9 {
10     int roll, flag=0;
11     struct Student s[5];
12     printf("Enter the informations of students\n-----\n");
13     for(int i=0; i<5; i++)
14     {
15         printf("> Student no - %d\n", i+1);
16         printf("Roll No. ");
17         scanf("%d", &s[i].roll);
18         printf("Name: ");
19         scanf("%s", &s[i].name);
20         printf("Age: ");
21         scanf("%d", &s[i].age);
22     }
23
24     printf("\nEnter Roll no to find details: ");
25     scanf("%d", &roll);
26     for(int i=0; i<5; i++)
27     {
28         if(s[i].roll == roll)
29         {
30             printf("Information of Roll No: %d\n-----\n", roll);
31             printf("Roll No: %d\nName: %s\nAge: %d\n", s[i].roll, s[i].name, s[i].age);
32             flag=1;
33             break;
34         }
35     }
36
37     if(flag==0)
38     {
39         printf("Student not found\n");
40     }
41     return 0;
42 }
```

The execution output shows the program running and displaying the details for the student with roll number 5:

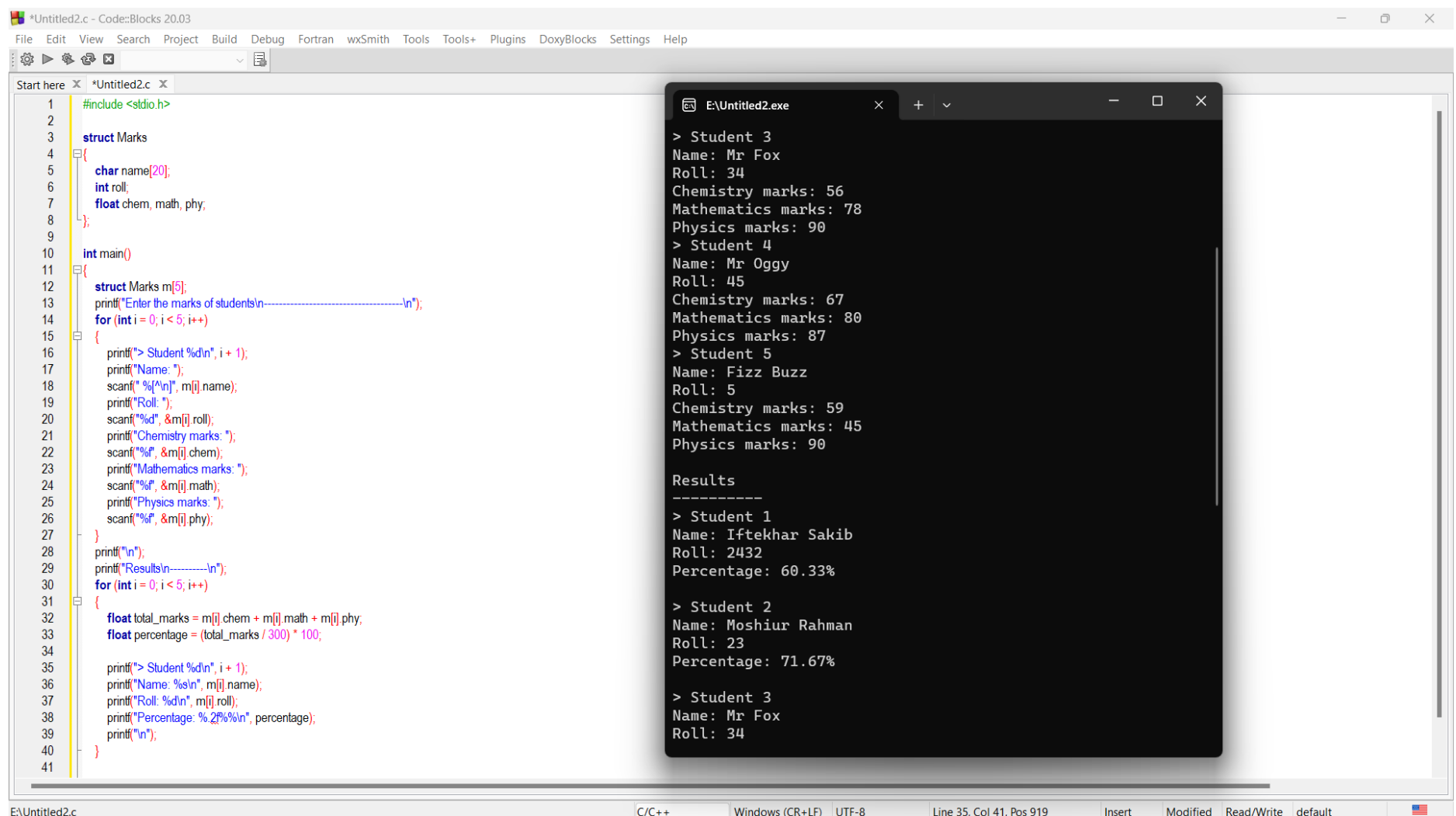
```
E:\Untitled1.exe
Enter the informations of students
-----
> Student no - 1
Roll No: 21
Name: Mr Fox
Age: 21
> Student no - 2
Roll No: 5
Name: Mr Dumpty
Age: 24
> Student no - 3
Roll No: 10
Name: Mr Humpty
Age: 10
> Student no - 4
Roll No: 15
Name: Oggy Foggy
Age: 18
> Student no - 5
Roll No: 4
Name: John Don
Age: 21

Enter Roll no to find details: 5
Information of Roll No: 5
-----
Roll No: 5
Name: Mr Dumpty
Age: 24

Process returned 0 (0x0)   execution time : 94.596 s
```

Problem no – 6

Enter the marks of 5 students in Chemistry, Mathematics and Physics (each out of 100) using a structure named Marks having elements roll no., name, chem_marks, maths_marks and phy_marks and then display the percentage of each student.



The screenshot shows the Code::Blocks IDE with a C program for Problem 6. The code defines a `Marks` structure with `name`, `roll`, `chem`, `math`, and `phy` fields. It uses a loop to input data for 5 students and another loop to calculate and display the percentage for each student.

```
1 #include <stdio.h>
2
3 struct Marks
4 {
5     char name[20];
6     int roll;
7     float chem, math, phy;
8 }
9
10 int main()
11 {
12     struct Marks m[5];
13     printf("Enter the marks of students\n-----\n");
14     for (int i = 0; i < 5; i++)
15     {
16         printf("> Student %d\n", i + 1);
17         printf("Name: ");
18         scanf("%s", m[i].name);
19         printf("Roll: ");
20         scanf("%d", &m[i].roll);
21         printf("Chemistry marks: ");
22         scanf("%f", &m[i].chem);
23         printf("Mathematics marks: ");
24         scanf("%f", &m[i].math);
25         printf("Physics marks: ");
26         scanf("%f", &m[i].phy);
27     }
28     printf("\n");
29     printf("Results\n-----\n");
30     for (int i = 0; i < 5; i++)
31     {
32         float total_marks = m[i].chem + m[i].math + m[i].phy;
33         float percentage = (total_marks / 300) * 100;
34
35         printf("> Student %d\n", i + 1);
36         printf("Name: %s\n", m[i].name);
37         printf("Roll: %d\n", m[i].roll);
38         printf("Percentage: %2f%%\n", percentage);
39         printf("\n");
40     }
41 }
```

The execution output shows the program running and displaying the percentage for each student:

```
E:\Untitled2.exe
> Student 3
Name: Mr Fox
Roll: 34
Chemistry marks: 56
Mathematics marks: 78
Physics marks: 90
> Student 4
Name: Mr Oggy
Roll: 45
Chemistry marks: 67
Mathematics marks: 80
Physics marks: 87
> Student 5
Name: Fizz Buzz
Roll: 5
Chemistry marks: 59
Mathematics marks: 45
Physics marks: 90

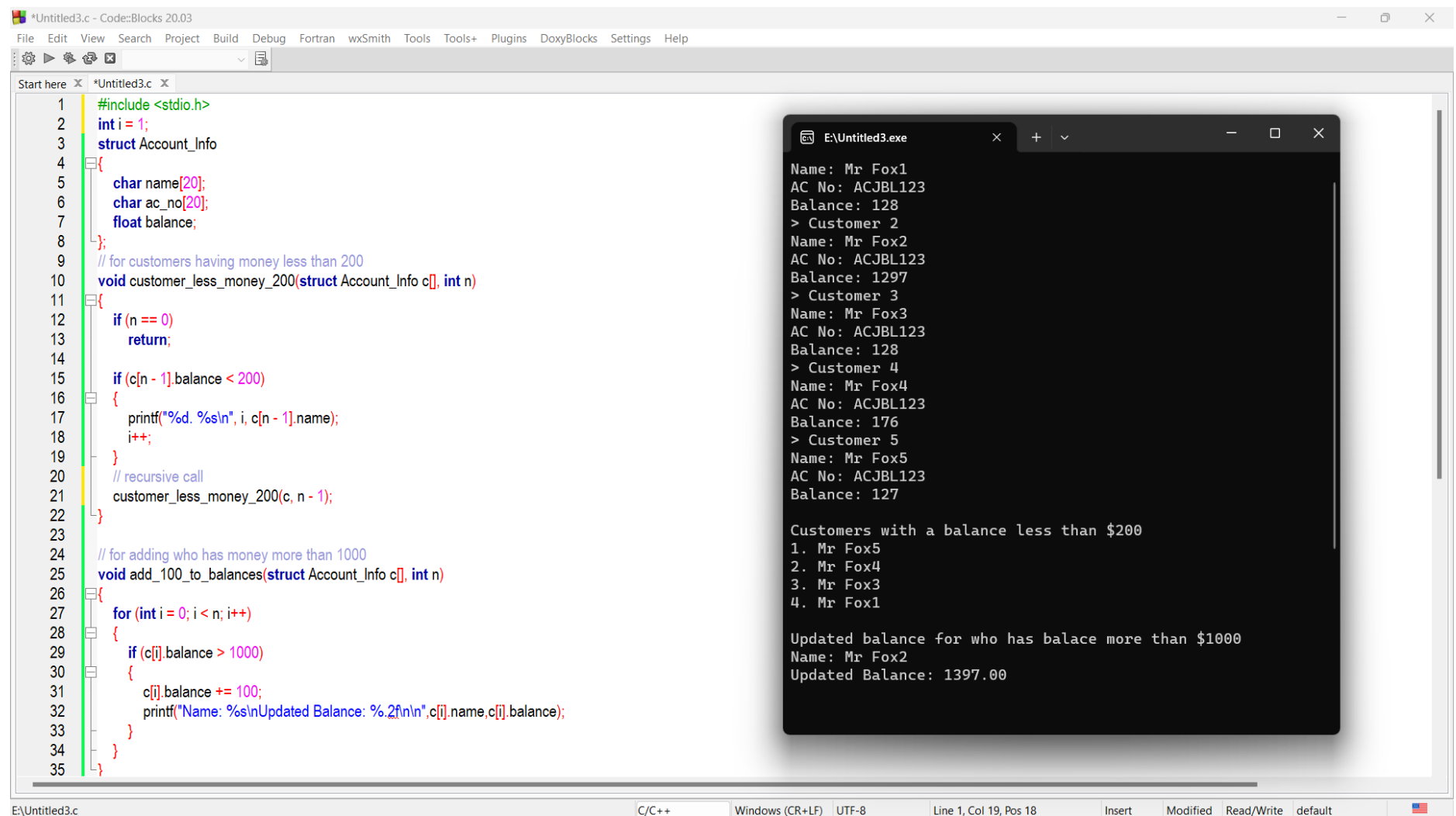
Results
-----
> Student 1
Name: Iftekhar Sakib
Roll: 2432
Percentage: 60.33%

> Student 2
Name: Moshiur Rahman
Roll: 23
Percentage: 71.67%

> Student 3
Name: Mr Fox
Roll: 34
```

Problem no – 7

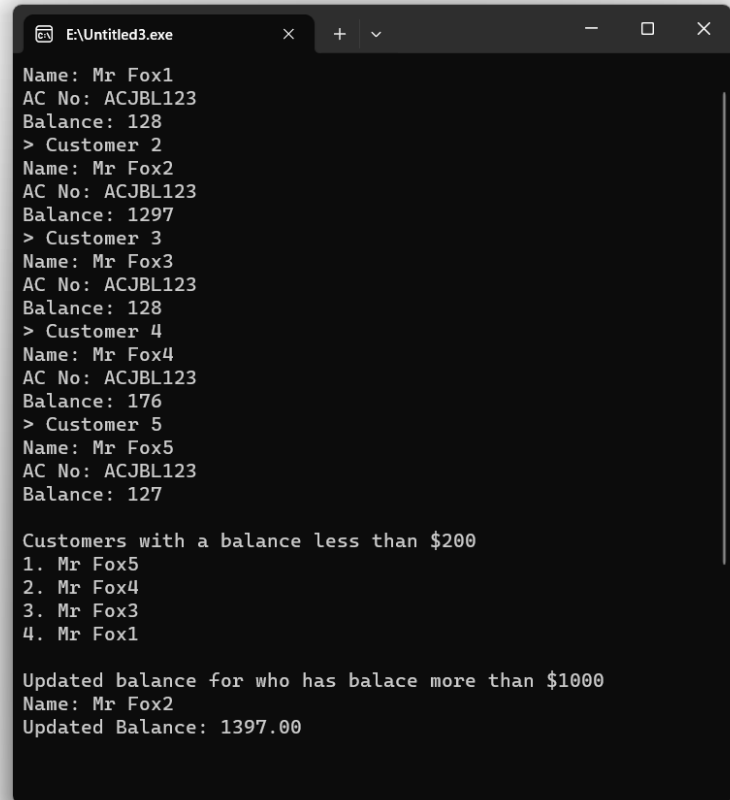
Write a structure to store the name, account number and balance of customers (more than 10) and store their information.



```
1 #include <stdio.h>
2 int i = 1;
3 struct Account_Info
4 {
5     char name[20];
6     char ac_no[20];
7     float balance;
8 };
9 // for customers having money less than 200
10 void customer_less_money_200(struct Account_Info c[], int n)
11 {
12     if (n == 0)
13         return;
14     if (c[n - 1].balance < 200)
15     {
16         printf("%d. %s\n", i, c[n - 1].name);
17         i++;
18     }
19     // recursive call
20     customer_less_money_200(c, n - 1);
21 }
22
23 // for adding who has money more than 1000
24 void add_100_to_balances(struct Account_Info c[], int n)
25 {
26     for (int i = 0; i < n; i++)
27     {
28         if (c[i].balance > 1000)
29         {
30             c[i].balance += 100;
31             printf("Name: %s\nUpdated Balance: %.2f\n", c[i].name, c[i].balance);
32         }
33     }
34 }
35
```

E:\Untitled3.c

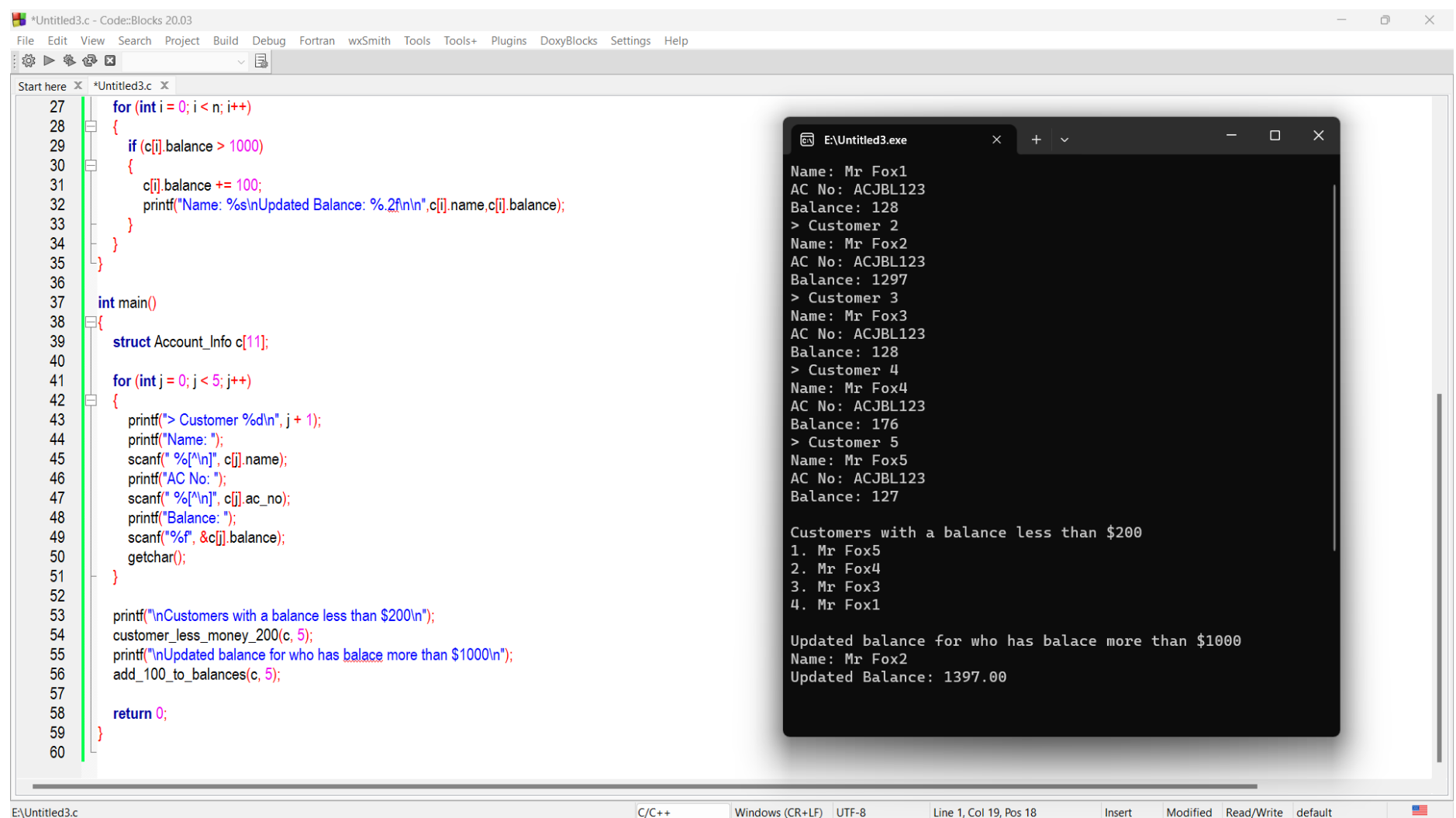
C/C++ Windows (CR+LF) UTF-8 Line 1, Col 19, Pos 18 Insert Modified Read/Write default



```
E:\Untitled3.exe
Name: Mr Fox1
AC No: ACJBL123
Balance: 128
> Customer 2
Name: Mr Fox2
AC No: ACJBL123
Balance: 1297
> Customer 3
Name: Mr Fox3
AC No: ACJBL123
Balance: 128
> Customer 4
Name: Mr Fox4
AC No: ACJBL123
Balance: 176
> Customer 5
Name: Mr Fox5
AC No: ACJBL123
Balance: 127

Customers with a balance less than $200
1. Mr Fox5
2. Mr Fox4
3. Mr Fox3
4. Mr Fox1

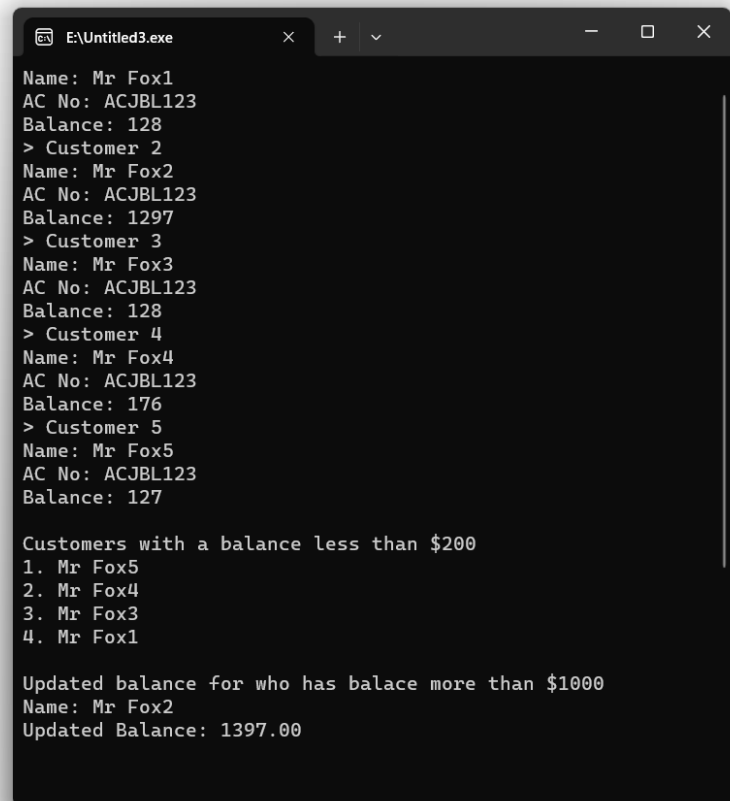
Updated balance for who has balance more than $1000
Name: Mr Fox2
Updated Balance: 1397.00
```



```
27 for (int i = 0; i < n; i++)
28 {
29     if (c[i].balance > 1000)
30     {
31         c[i].balance += 100;
32         printf("Name: %s\nUpdated Balance: %.2f\n", c[i].name, c[i].balance);
33     }
34 }
35
36 int main()
37 {
38     struct Account_Info c[11];
39     for (int j = 0; j < 5; j++)
40     {
41         printf("> Customer %d\n", j + 1);
42         printf("Name: ");
43         scanf("%s", c[j].name);
44         printf("AC No: ");
45         scanf("%s", c[j].ac_no);
46         printf("Balance: ");
47         scanf("%f", &c[j].balance);
48         getchar();
49     }
50
51     printf("\nCustomers with a balance less than $200\n");
52     customer_less_money_200(c, 5);
53     printf("\nUpdated balance for who has balance more than $1000\n");
54     add_100_to_balances(c, 5);
55
56     return 0;
57 }
58
59
60
```

E:\Untitled3.c

C/C++ Windows (CR+LF) UTF-8 Line 1, Col 19, Pos 18 Insert Modified Read/Write default



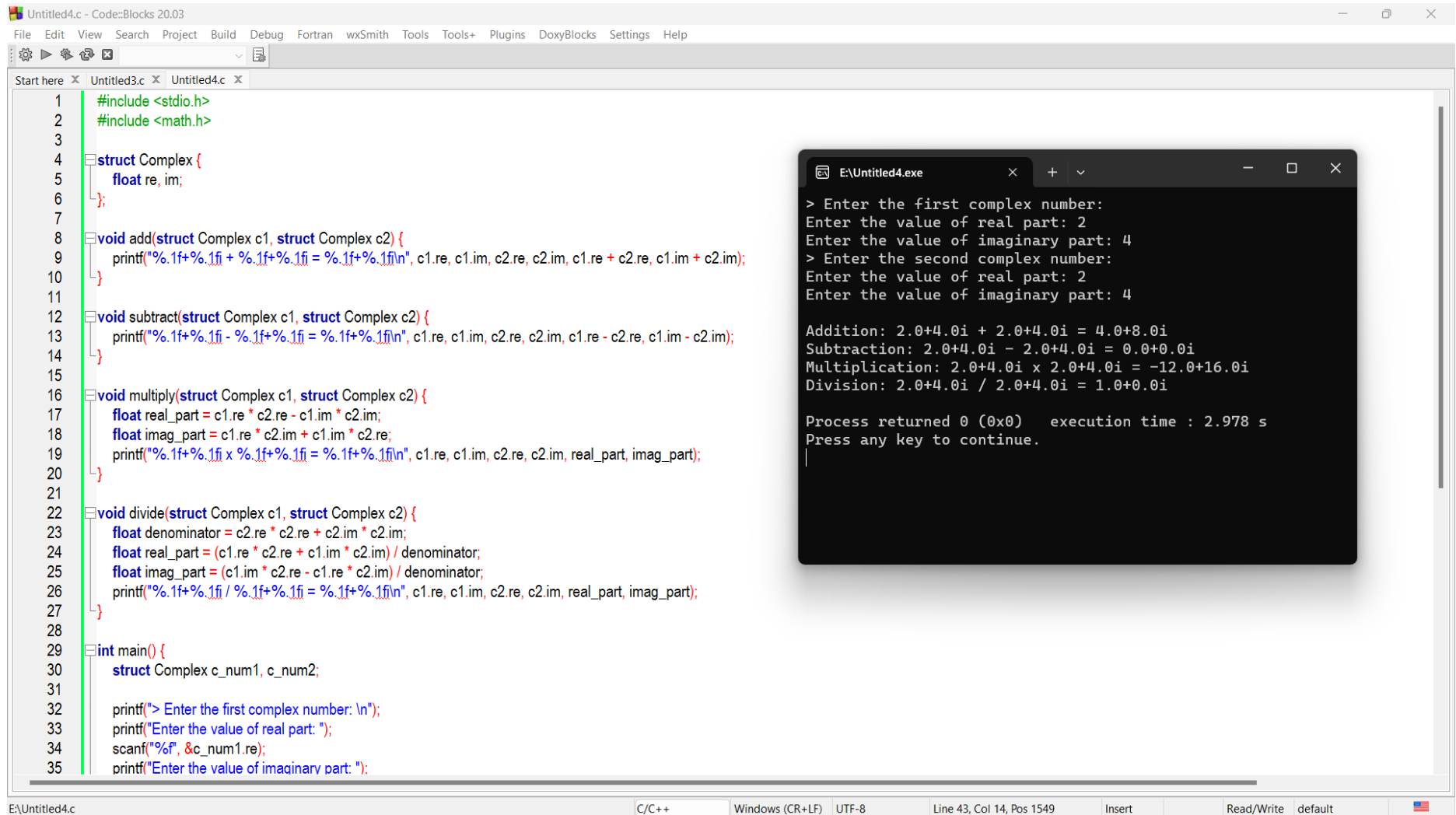
```
E:\Untitled3.exe
Name: Mr Fox1
AC No: ACJBL123
Balance: 128
> Customer 2
Name: Mr Fox2
AC No: ACJBL123
Balance: 1297
> Customer 3
Name: Mr Fox3
AC No: ACJBL123
Balance: 128
> Customer 4
Name: Mr Fox4
AC No: ACJBL123
Balance: 176
> Customer 5
Name: Mr Fox5
AC No: ACJBL123
Balance: 127

Customers with a balance less than $200
1. Mr Fox5
2. Mr Fox4
3. Mr Fox3
4. Mr Fox1

Updated balance for who has balance more than $1000
Name: Mr Fox2
Updated Balance: 1397.00
```

Problem no – 8

Write a program to add, subtract and multiply two complex numbers using structures to function (use function for each operation).



The screenshot shows the Code::Blocks IDE with a C program for complex number operations. The program defines a `Complex` structure and functions for addition, subtraction, multiplication, and division. The `main` function prompts the user for two complex numbers and calls these functions. The execution window shows the program running with input values 2 and 4 for both real and imaginary parts of the two numbers. The output displays the results of addition, subtraction, multiplication, and division, along with the execution time and a prompt to press any key to continue.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 struct Complex {
5     float re, im;
6 };
7
8 void add(struct Complex c1, struct Complex c2) {
9     printf("%.1f+%.1fi + %.1f+%.1fi = %.1f+%.1fi\n", c1.re, c1.im, c2.re, c2.im, c1.re + c2.re, c1.im + c2.im);
10 }
11
12 void subtract(struct Complex c1, struct Complex c2) {
13     printf("%.1f+%.1fi - %.1f+%.1fi = %.1f+%.1fi\n", c1.re, c1.im, c2.re, c2.im, c1.re - c2.re, c1.im - c2.im);
14 }
15
16 void multiply(struct Complex c1, struct Complex c2) {
17     float real_part = c1.re * c2.re - c1.im * c2.im;
18     float imag_part = c1.re * c2.im + c1.im * c2.re;
19     printf("%.1f+%.1fi x %.1f+%.1fi = %.1f+%.1fi\n", c1.re, c1.im, c2.re, c2.im, real_part, imag_part);
20 }
21
22 void divide(struct Complex c1, struct Complex c2) {
23     float denominator = c2.re * c2.re + c2.im * c2.im;
24     float real_part = (c1.re * c2.re + c1.im * c2.im) / denominator;
25     float imag_part = (c1.im * c2.re - c1.re * c2.im) / denominator;
26     printf("%.1f+%.1fi / %.1f+%.1fi = %.1f+%.1fi\n", c1.re, c1.im, c2.re, c2.im, real_part, imag_part);
27 }
28
29 int main() {
30     struct Complex c_num1, c_num2;
31
32     printf("> Enter the first complex number: \n");
33     printf("Enter the value of real part: ");
34     scanf("%f", &c_num1.re);
35     printf("Enter the value of imaginary part: ");
```

```
> Enter the first complex number:
Enter the value of real part: 2
Enter the value of imaginary part: 4
> Enter the second complex number:
Enter the value of real part: 2
Enter the value of imaginary part: 4

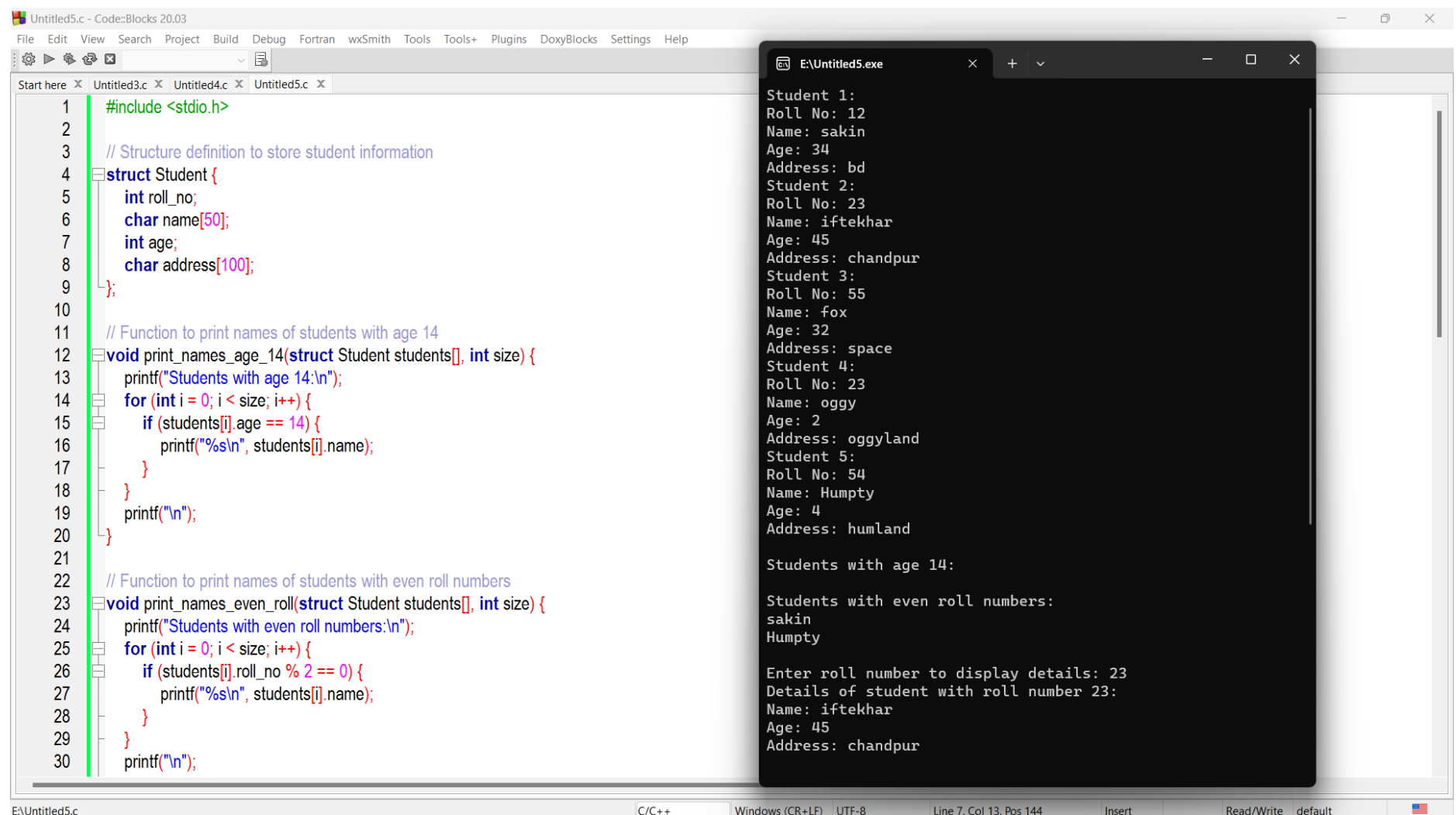
Addition: 2.0+4.0i + 2.0+4.0i = 4.0+8.0i
Subtraction: 2.0+4.0i - 2.0+4.0i = 0.0+0.0i
Multiplication: 2.0+4.0i x 2.0+4.0i = -12.0+16.0i
Division: 2.0+4.0i / 2.0+4.0i = 1.0+0.0i

Process returned 0 (0x0)   execution time : 2.978 s
Press any key to continue.
```

Problem no – 9

Write a structure to store the roll no., name, age (between 11 to 14) and address of students. Store the information of the students.

- Write a function to print the names of all the students having age 14.
- Write another function to print the names of all the students having even roll no.
- Write another function to display the details of the student whose roll no is given (i.e. roll no. entered by the user).



The screenshot displays the Code::Blocks IDE with a C++ program. The code defines a `Student` structure and three functions: `print_names_age_14`, `print_names_even_roll`, and a main function that uses them. The execution window shows the output of these functions, including student details for five students, names of students with age 14, names of students with even roll numbers, and details for a specific student (roll number 23).

```
#include <stdio.h>

// Structure definition to store student information
struct Student {
    int roll_no;
    char name[50];
    int age;
    char address[100];
};

// Function to print names of students with age 14
void print_names_age_14(struct Student students[], int size) {
    printf("Students with age 14:\n");
    for (int i = 0; i < size; i++) {
        if (students[i].age == 14) {
            printf("%s\n", students[i].name);
        }
    }
    printf("\n");
}

// Function to print names of students with even roll numbers
void print_names_even_roll(struct Student students[], int size) {
    printf("Students with even roll numbers:\n");
    for (int i = 0; i < size; i++) {
        if (students[i].roll_no % 2 == 0) {
            printf("%s\n", students[i].name);
        }
    }
    printf("\n");
}

int main() {
    struct Student students[5];
    // Student 1
    students[0].roll_no = 12;
    strcpy(students[0].name, "sakin");
    students[0].age = 34;
    strcpy(students[0].address, "bd");
    // Student 2
    students[1].roll_no = 23;
    strcpy(students[1].name, "iftekhhar");
    students[1].age = 45;
    strcpy(students[1].address, "chandpur");
    // Student 3
    students[2].roll_no = 55;
    strcpy(students[2].name, "fox");
    students[2].age = 32;
    strcpy(students[2].address, "space");
    // Student 4
    students[3].roll_no = 23;
    strcpy(students[3].name, "oggy");
    students[3].age = 2;
    strcpy(students[3].address, "oggyland");
    // Student 5
    students[4].roll_no = 54;
    strcpy(students[4].name, "Humpty");
    students[4].age = 4;
    strcpy(students[4].address, "humland");

    print_names_age_14(students, 5);
    print_names_even_roll(students, 5);

    int roll_no;
    printf("Enter roll number to display details: ");
    scanf("%d", &roll_no);

    for (int i = 0; i < 5; i++) {
        if (students[i].roll_no == roll_no) {
            printf("Details of student with roll number %d:\n", roll_no);
            printf("Name: %s\n", students[i].name);
            printf("Age: %d\n", students[i].age);
            printf("Address: %s\n", students[i].address);
        }
    }

    return 0;
}
```

Execution Output:

```
Student 1:
Roll No: 12
Name: sakin
Age: 34
Address: bd
Student 2:
Roll No: 23
Name: iftekhar
Age: 45
Address: chandpur
Student 3:
Roll No: 55
Name: fox
Age: 32
Address: space
Student 4:
Roll No: 23
Name: oggy
Age: 2
Address: oggyland
Student 5:
Roll No: 54
Name: Humpty
Age: 4
Address: humland

Students with age 14:

Students with even roll numbers:
sakin
Humpty

Enter roll number to display details: 23
Details of student with roll number 23:
Name: iftekhar
Age: 45
Address: chandpur
```

N.B. Problem number 9's code was taken from AI. Although I understand the algorithm, I couldn't write it due to academic pressure.