JAMIA MILLIA ISLAMIA

C LAB PROGRAMS

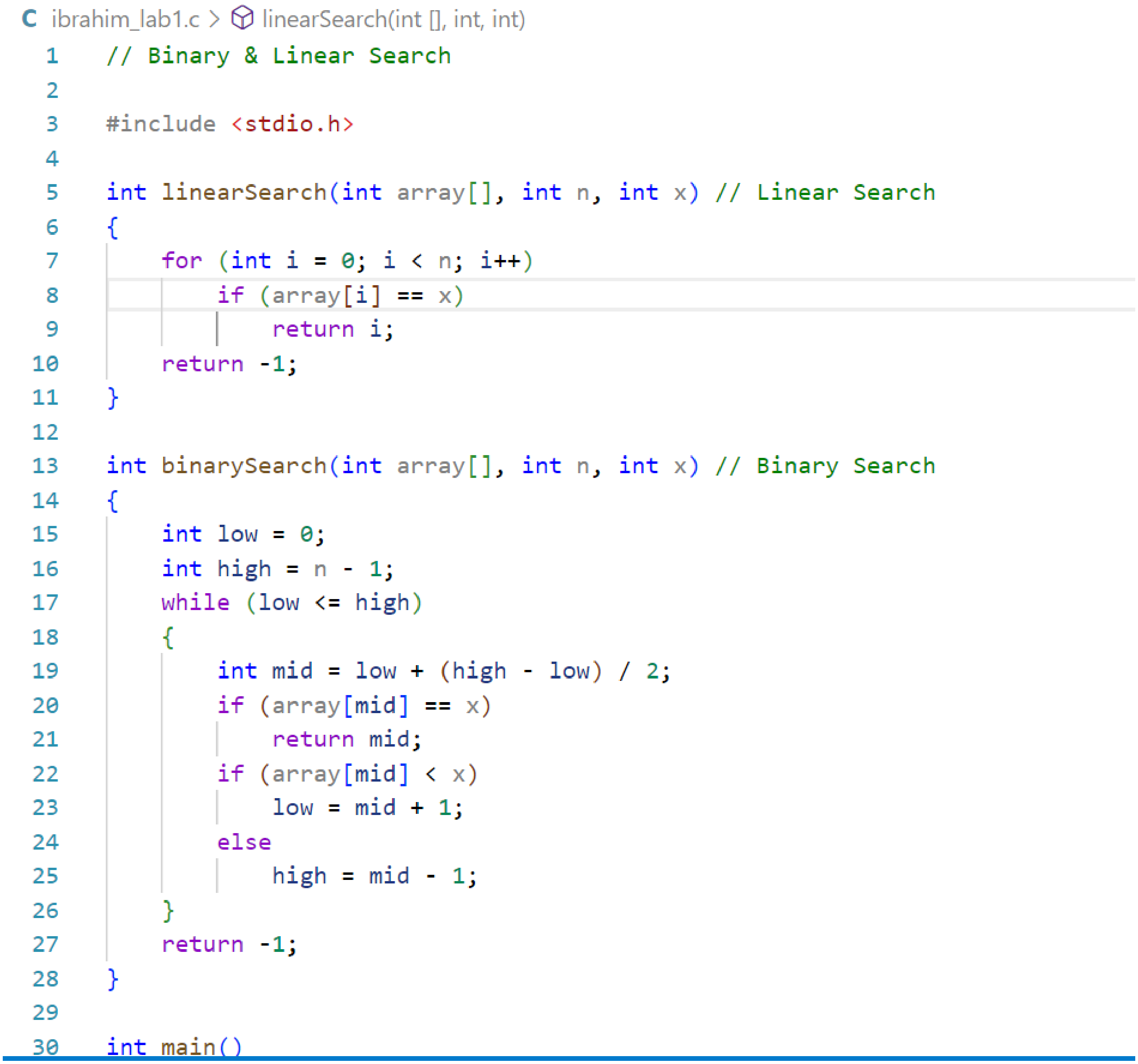
IIIRD SEMESTER

SESSION: 2022-2023

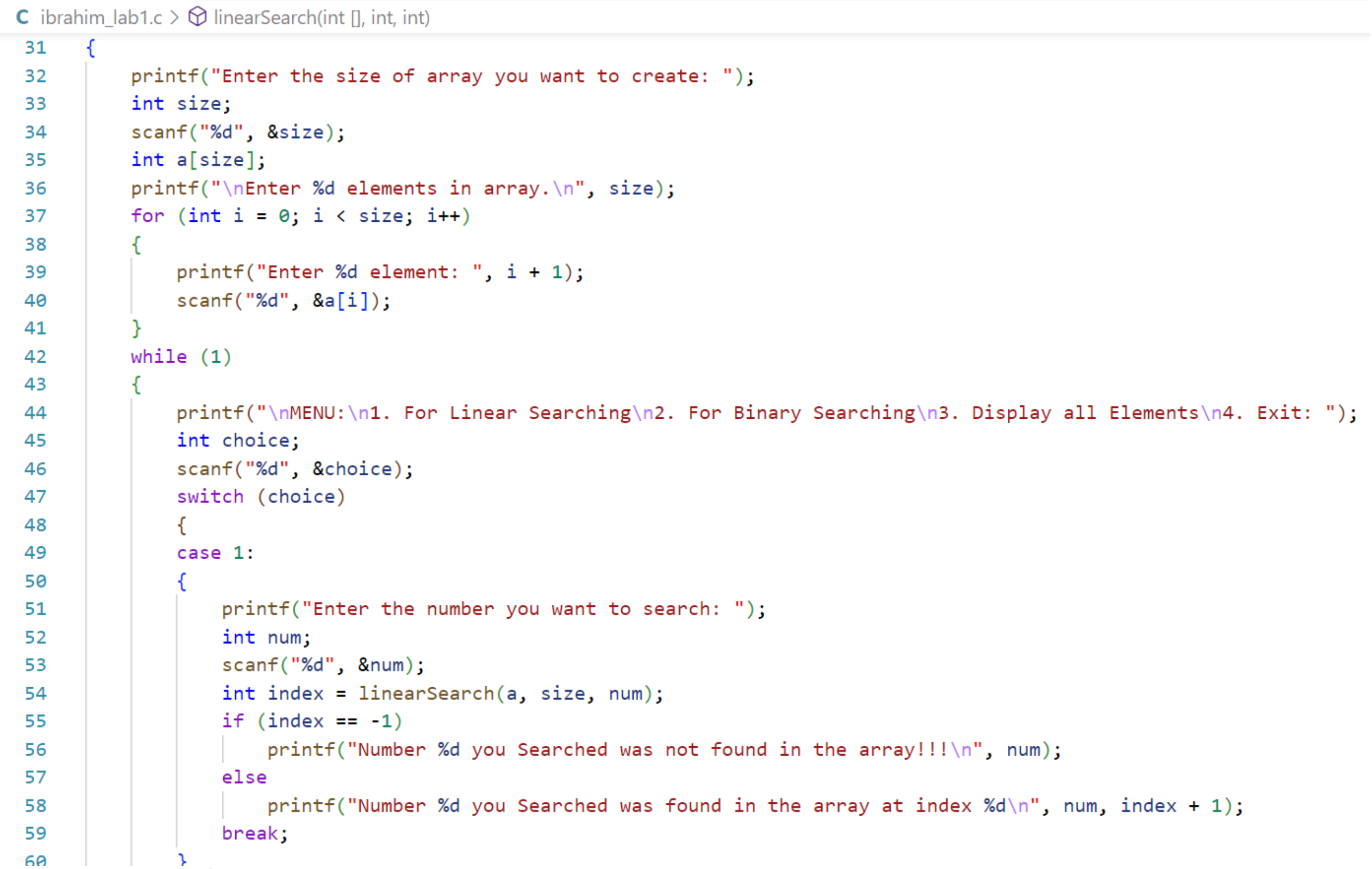
- **MD. IBRAHIM AKHTAR**
- **Dept. of Computer Science**
- **Roll No.: 21BCS007**
- **Serial No.: 07**
- **Semester: 3**

INDEX

Program: 1

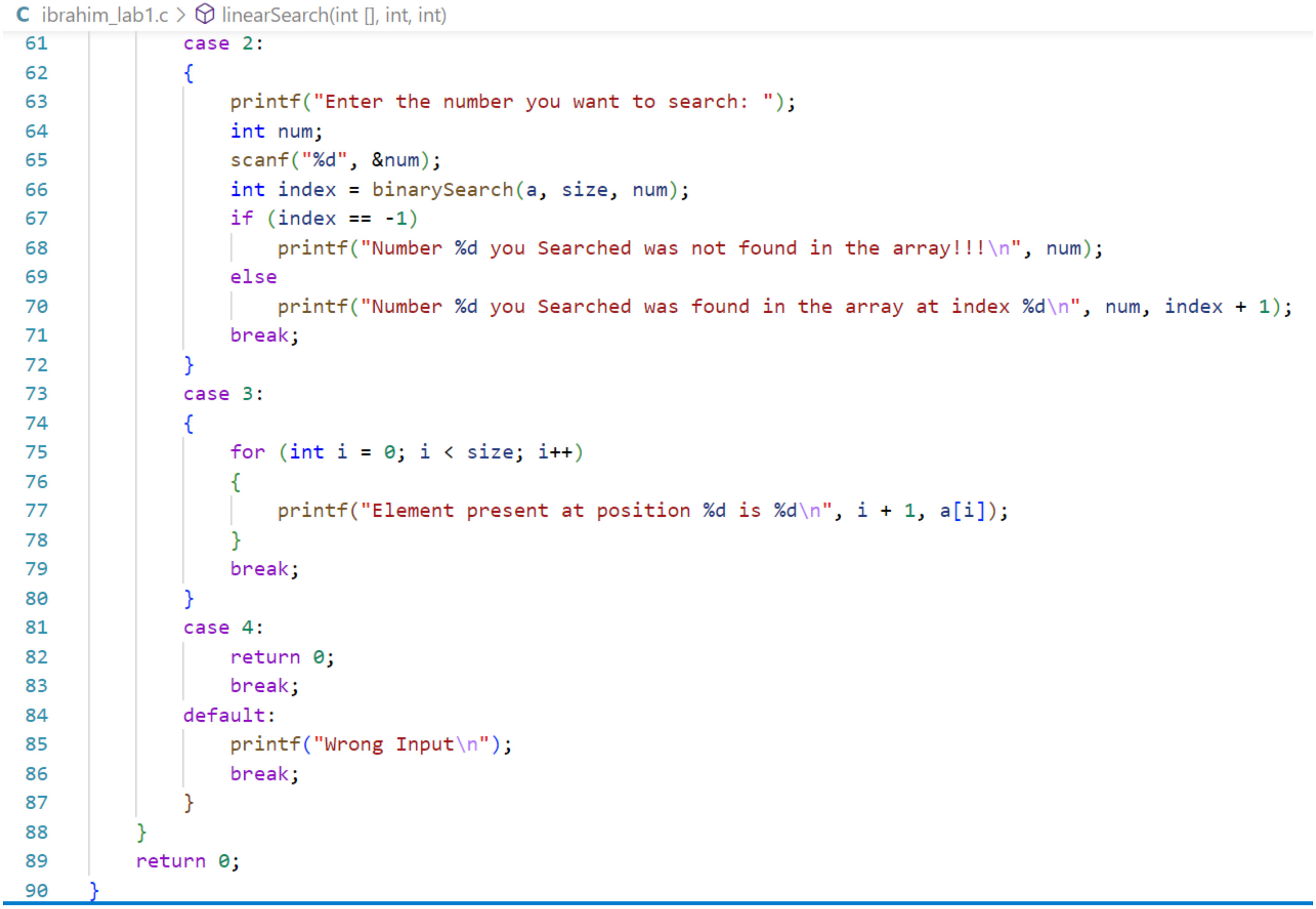


```c
// Binary & Linear Search

#include <stdio.h>

int linearSearch(int array[], int n, int x) // Linear Search
{
    for (int i = 0; i < n; i++)
        if (array[i] == x)
            return i;
    return -1;
}

int binarySearch(int array[], int n, int x) // Binary Search
{
    int low = 0;
    int high = n - 1;
    while (low <= high)
    {
        int mid = low + (high - low) / 2;
        if (array[mid] == x)
            return mid;
        if (array[mid] < x)
            low = mid + 1;
        else
            high = mid - 1;
    }
    return -1;
}

int main()
```



C ibrahim_lab1.c > linearSearch(int [], int, int)

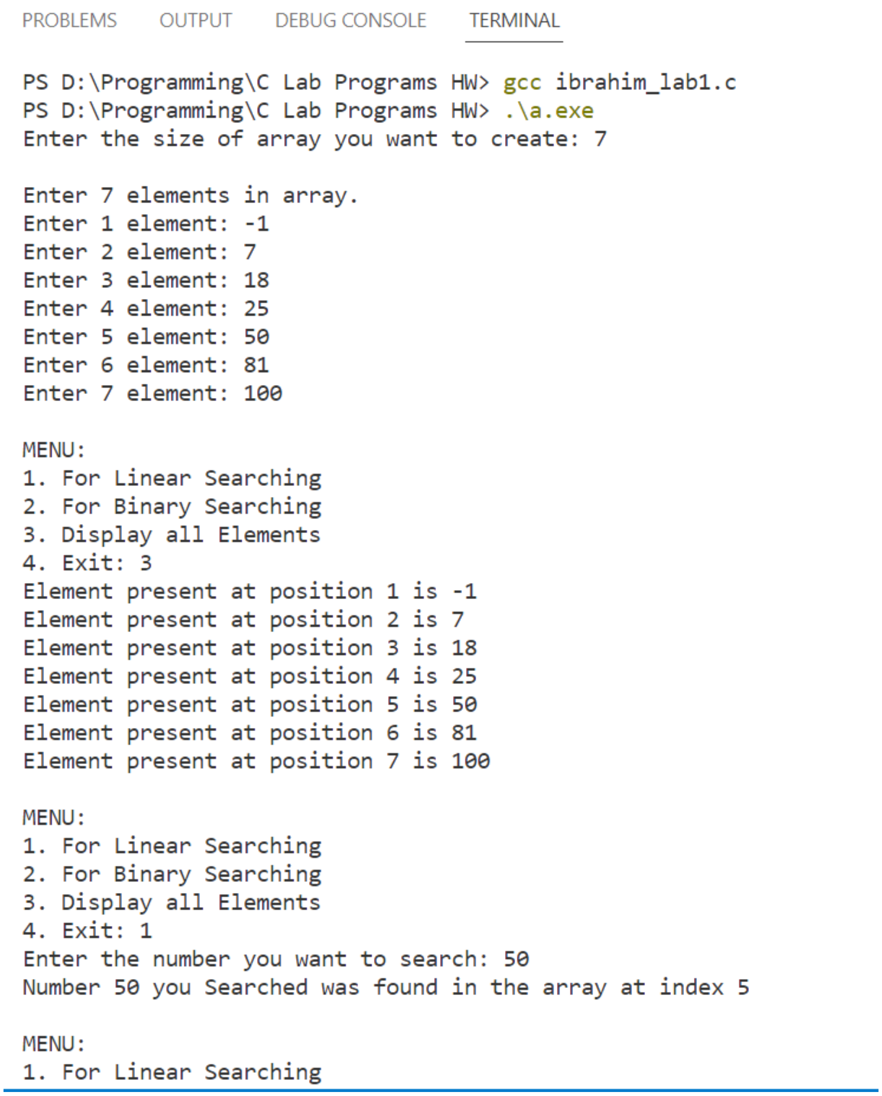
```c
{
    printf("Enter the size of array you want to create: ");
    int size;
    scanf("%d", &size);
    int a[size];
    printf("\nEnter %d elements in array.\n", size);
    for (int i = 0; i < size; i++)
    {
        printf("Enter %d element: ", i + 1);
        scanf("%d", &a[i]);
    }
    while (1)
    {
        printf("\nMENU:\n1. For Linear Searching\n2. For Binary Searching\n3. Display all Elements\n4. Exit: ");
        int choice;
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
        {
            printf("Enter the number you want to search: ");
            int num;
            scanf("%d", &num);
            int index = linearSearch(a, size, num);
            if (index == -1)
                printf("Number %d you Searched was not found in the array!!!\n", num);
            else
                printf("Number %d you Searched was found in the array at index %d\n", num, index + 1);
            break;
        }
```



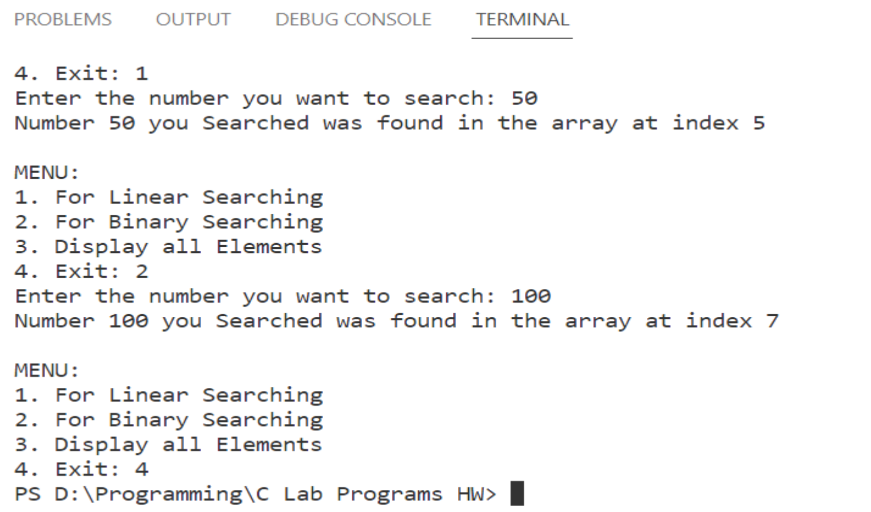
C ibrahim_lab1.c > linearSearch(int [], int, int)

```c
        case 2:
        {
            printf("Enter the number you want to search: ");
            int num;
            scanf("%d", &num);
            int index = binarySearch(a, size, num);
            if (index == -1)
                printf("Number %d you Searched was not found in the array!!!\n", num);
            else
                printf("Number %d you Searched was found in the array at index %d\n", num, index + 1);
            break;
        }
        case 3:
        {
            for (int i = 0; i < size; i++)
            {
                printf("Element present at position %d is %d\n", i + 1, a[i]);
            }
            break;
        }
        case 4:
            return 0;
            break;
        default:
            printf("Wrong Input\n");
            break;
        }
    }
    return 0;
}
```

Output: 1

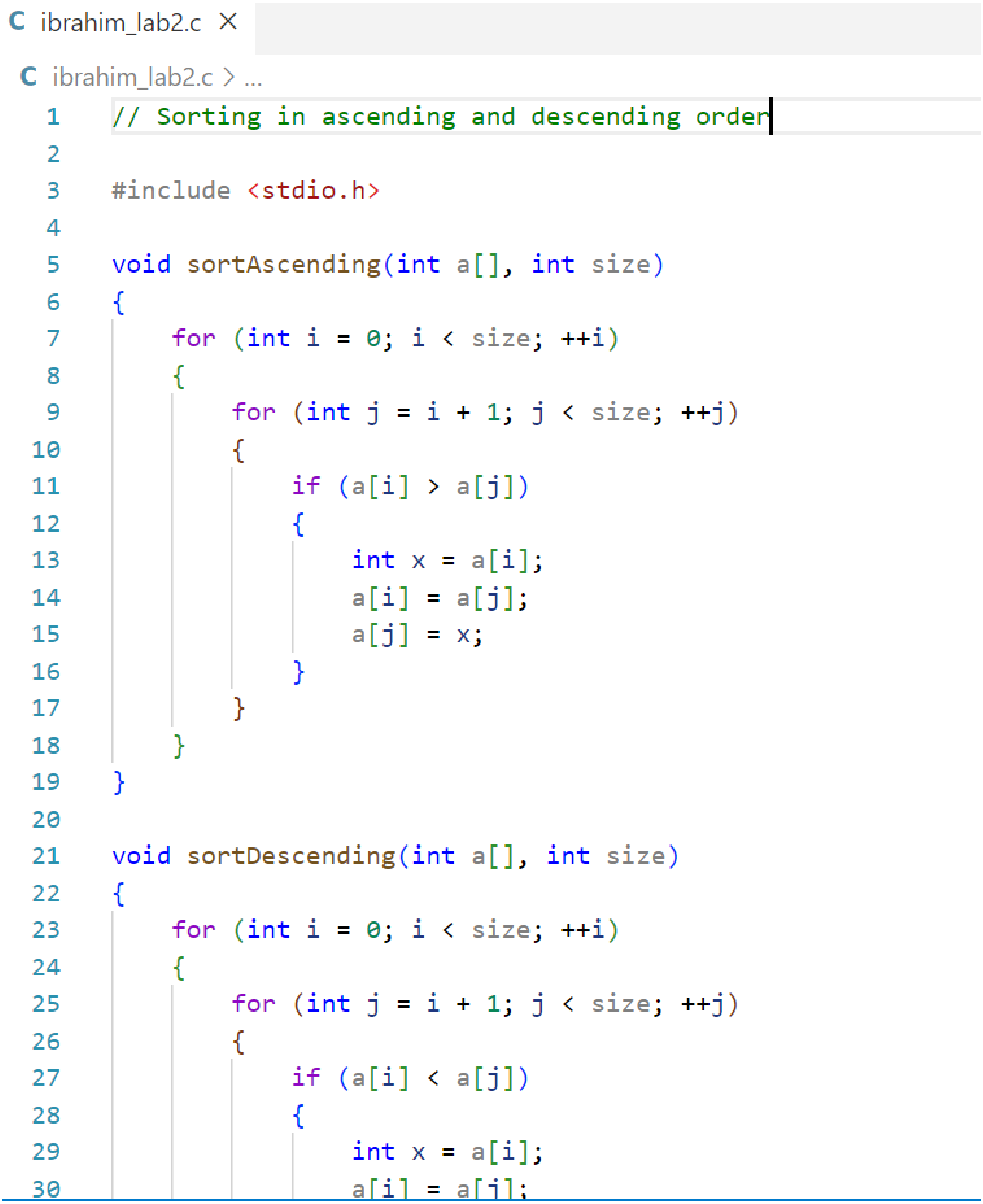


```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab1.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter the size of array you want to create: 7

Enter 7 elements in array.
Enter 1 element: -1
Enter 2 element: 7
Enter 3 element: 18
Enter 4 element: 25
Enter 5 element: 50
Enter 6 element: 81
Enter 7 element: 100

MENU:
1. For Linear Searching
2. For Binary Searching
3. Display all Elements
4. Exit: 3
Element present at position 1 is -1
Element present at position 2 is 7
Element present at position 3 is 18
Element present at position 4 is 25
Element present at position 5 is 50
Element present at position 6 is 81
Element present at position 7 is 100

MENU:
1. For Linear Searching
2. For Binary Searching
3. Display all Elements
4. Exit: 1
Enter the number you want to search: 50
Number 50 you Searched was found in the array at index 5

MENU:
1. For Linear Searching
```

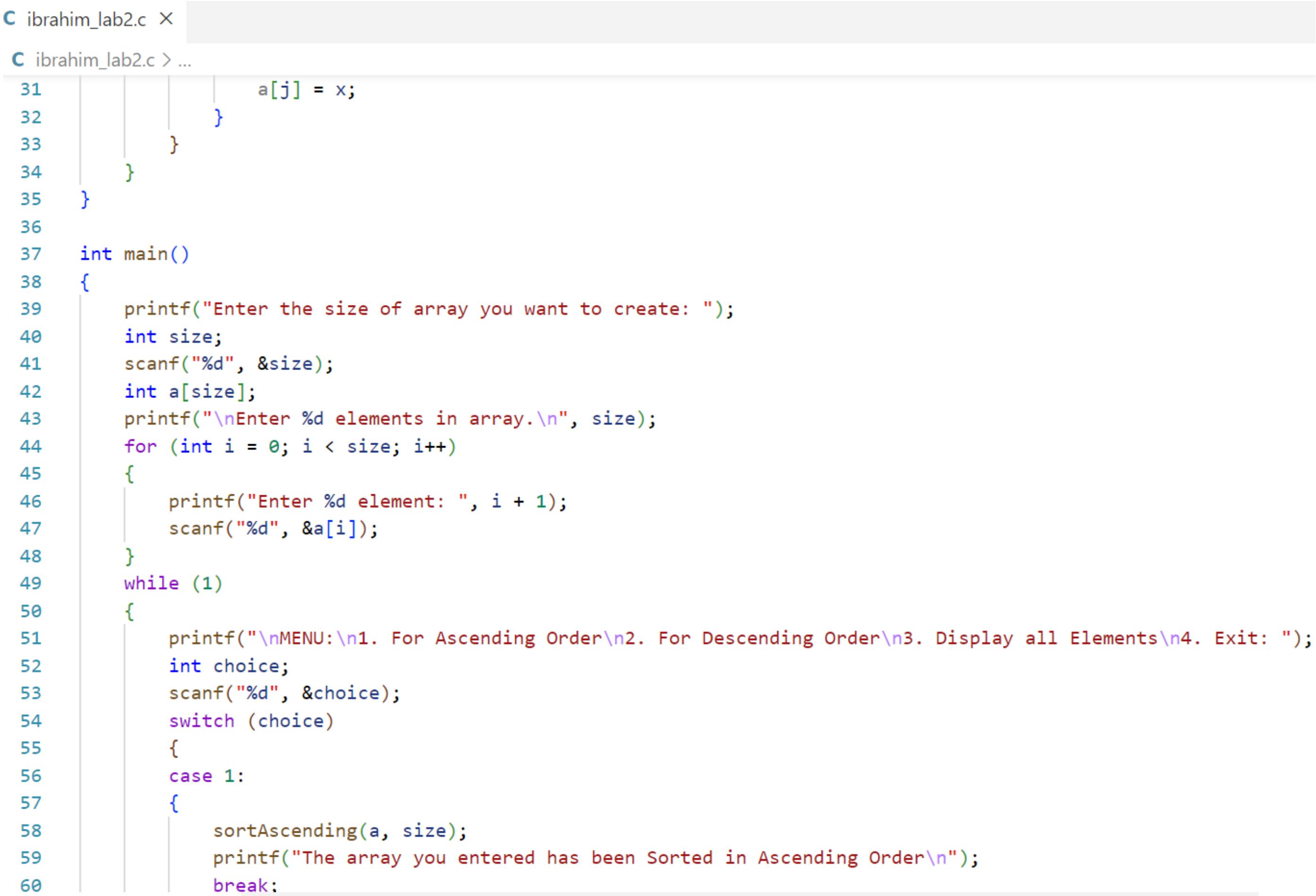


```
4. Exit: 1
Enter the number you want to search: 50
Number 50 you Searched was found in the array at index 5

MENU:
1. For Linear Searching
2. For Binary Searching
3. Display all Elements
4. Exit: 2
Enter the number you want to search: 100
Number 100 you Searched was found in the array at index 7

MENU:
1. For Linear Searching
2. For Binary Searching
3. Display all Elements
4. Exit: 4
PS D:\Programming\C Lab Programs HW> █
```

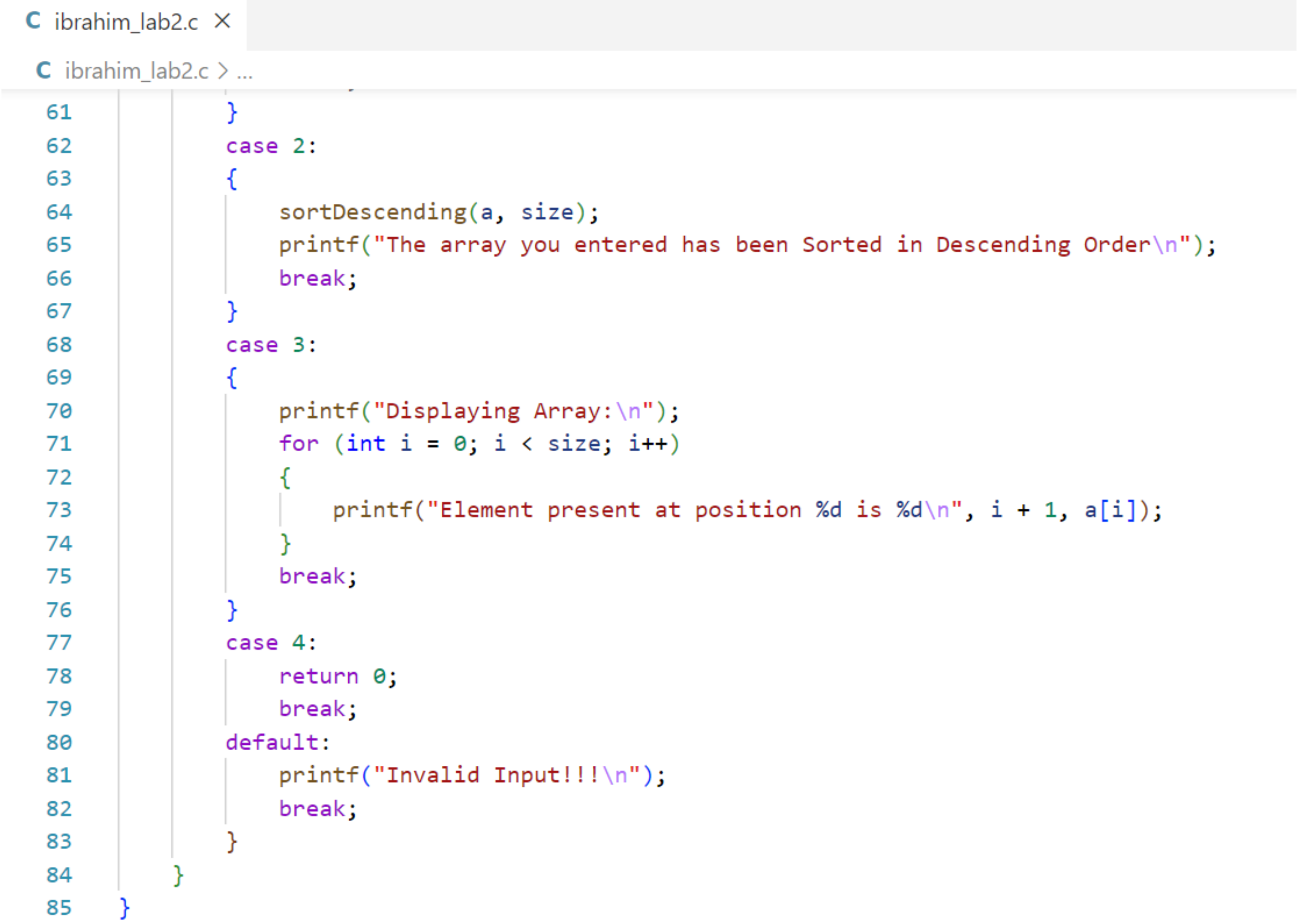
Program: 2



```c
// Sorting in ascending and descending order

#include <stdio.h>

void sortAscending(int a[], int size)
{
    for (int i = 0; i < size; ++i)
    {
        for (int j = i + 1; j < size; ++j)
        {
            if (a[i] > a[j])
            {
                int x = a[i];
                a[i] = a[j];
                a[j] = x;
            }
        }
    }
}

void sortDescending(int a[], int size)
{
    for (int i = 0; i < size; ++i)
    {
        for (int j = i + 1; j < size; ++j)
        {
            if (a[i] < a[j])
            {
                int x = a[i];
                a[i] = a[j];
```

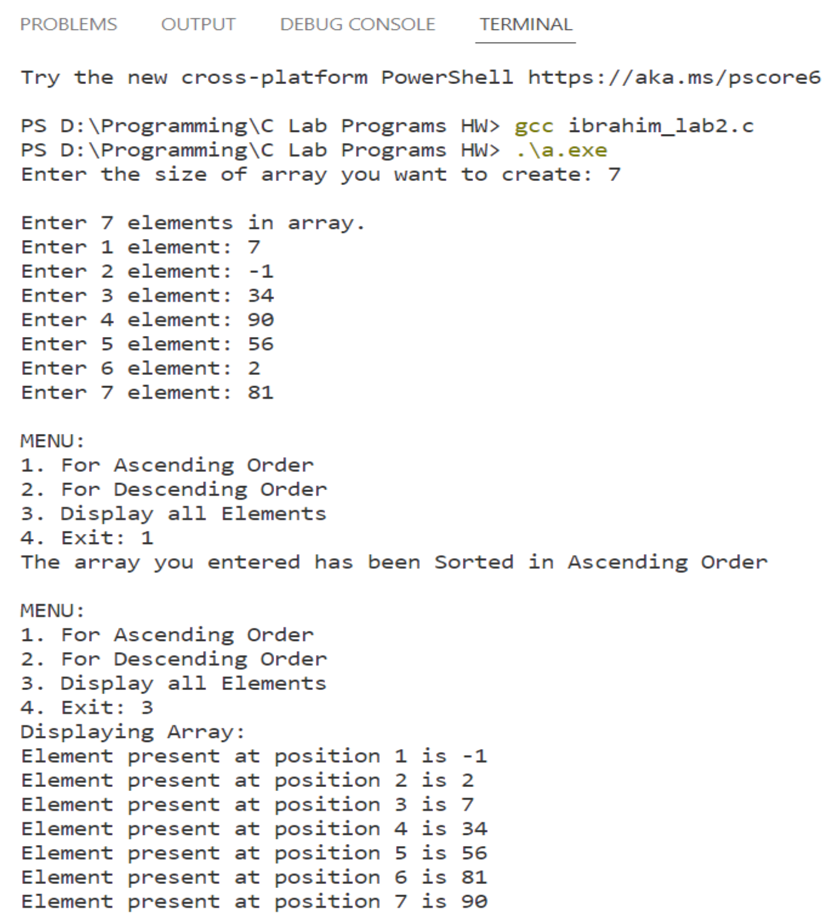


```c
                a[j] = x;
            }
        }
    }
}

int main()
{
    printf("Enter the size of array you want to create: ");
    int size;
    scanf("%d", &size);
    int a[size];
    printf("\nEnter %d elements in array.\n", size);
    for (int i = 0; i < size; i++)
    {
        printf("Enter %d element: ", i + 1);
        scanf("%d", &a[i]);
    }
    while (1)
    {
        printf("\nMENU:\n1. For Ascending Order\n2. For Descending Order\n3. Display all Elements\n4. Exit: ");
        int choice;
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
        {
            sortAscending(a, size);
            printf("The array you entered has been Sorted in Ascending Order\n");
            break;
```

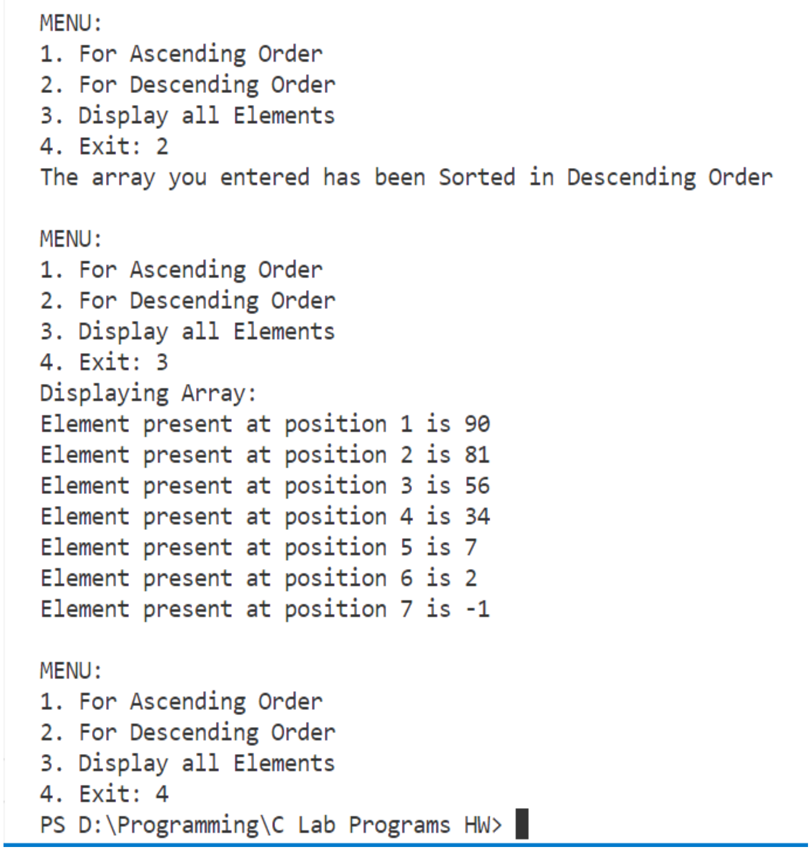


```c
        }
        case 2:
        {
            sortDescending(a, size);
            printf("The array you entered has been Sorted in Descending Order\n");
            break;
        }
        case 3:
        {
            printf("Displaying Array:\n");
            for (int i = 0; i < size; i++)
            {
                printf("Element present at position %d is %d\n", i + 1, a[i]);
            }
            break;
        }
        case 4:
            return 0;
            break;
        default:
            printf("Invalid Input!!!\n");
            break;
        }
    }
}
```

Output: 2

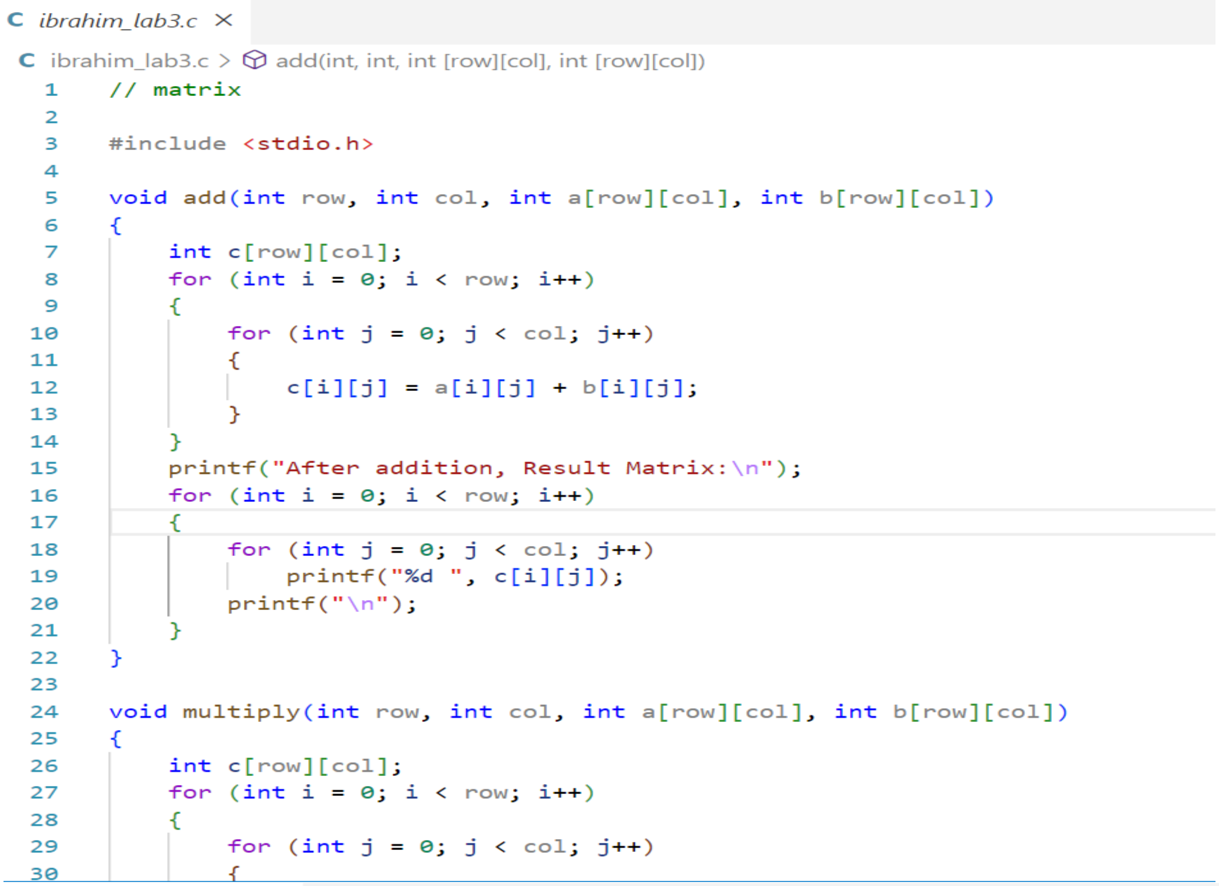


```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab2.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter the size of array you want to create: 7

Enter 7 elements in array.
Enter 1 element: 7
Enter 2 element: -1
Enter 3 element: 34
Enter 4 element: 90
Enter 5 element: 56
Enter 6 element: 2
Enter 7 element: 81

MENU:
1. For Ascending Order
2. For Descending Order
3. Display all Elements
4. Exit: 1
The array you entered has been Sorted in Ascending Order

MENU:
1. For Ascending Order
2. For Descending Order
3. Display all Elements
4. Exit: 3
Displaying Array:
Element present at position 1 is -1
Element present at position 2 is 2
Element present at position 3 is 7
Element present at position 4 is 34
Element present at position 5 is 56
Element present at position 6 is 81
Element present at position 7 is 90
```

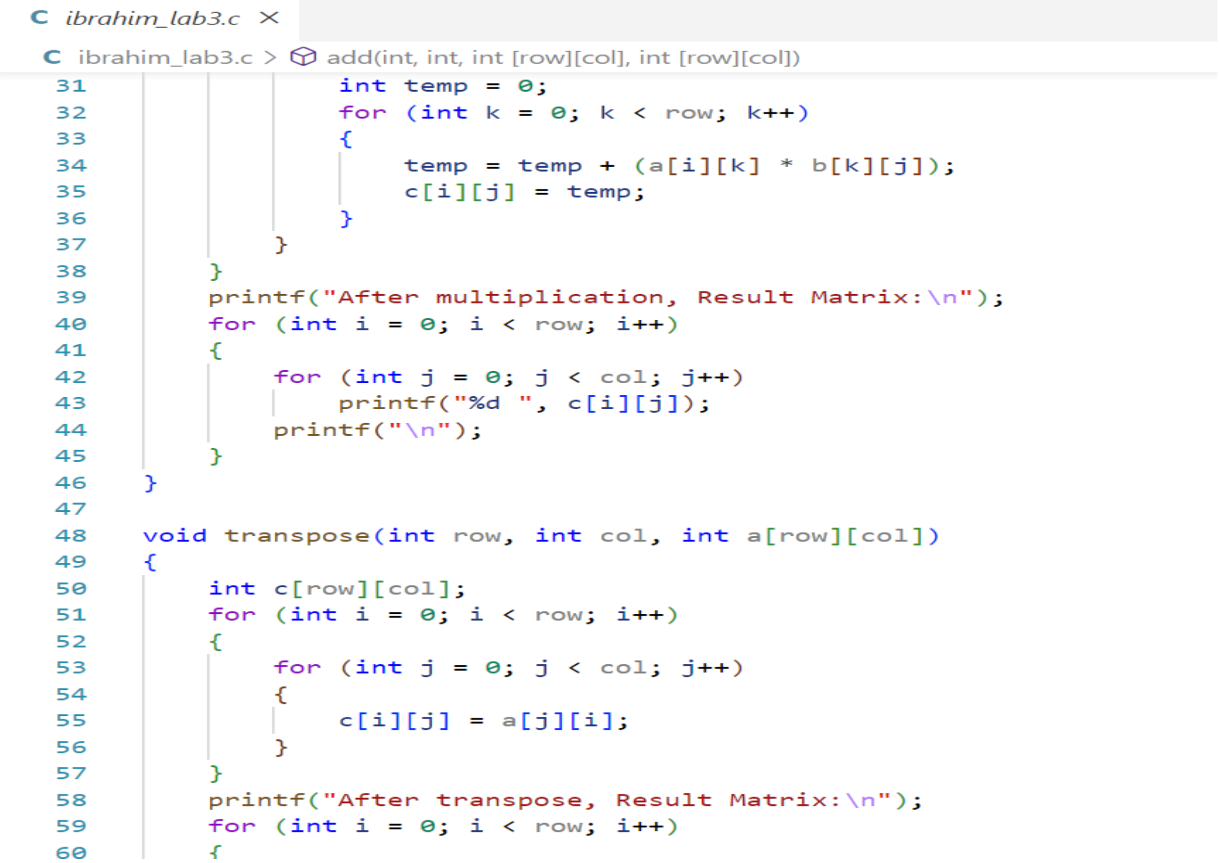


```
MENU:
1. For Ascending Order
2. For Descending Order
3. Display all Elements
4. Exit: 2
The array you entered has been Sorted in Descending Order

MENU:
1. For Ascending Order
2. For Descending Order
3. Display all Elements
4. Exit: 3
Displaying Array:
Element present at position 1 is 90
Element present at position 2 is 81
Element present at position 3 is 56
Element present at position 4 is 34
Element present at position 5 is 7
Element present at position 6 is 2
Element present at position 7 is -1

MENU:
1. For Ascending Order
2. For Descending Order
3. Display all Elements
4. Exit: 4
PS D:\Programming\C Lab Programs HW> 
```

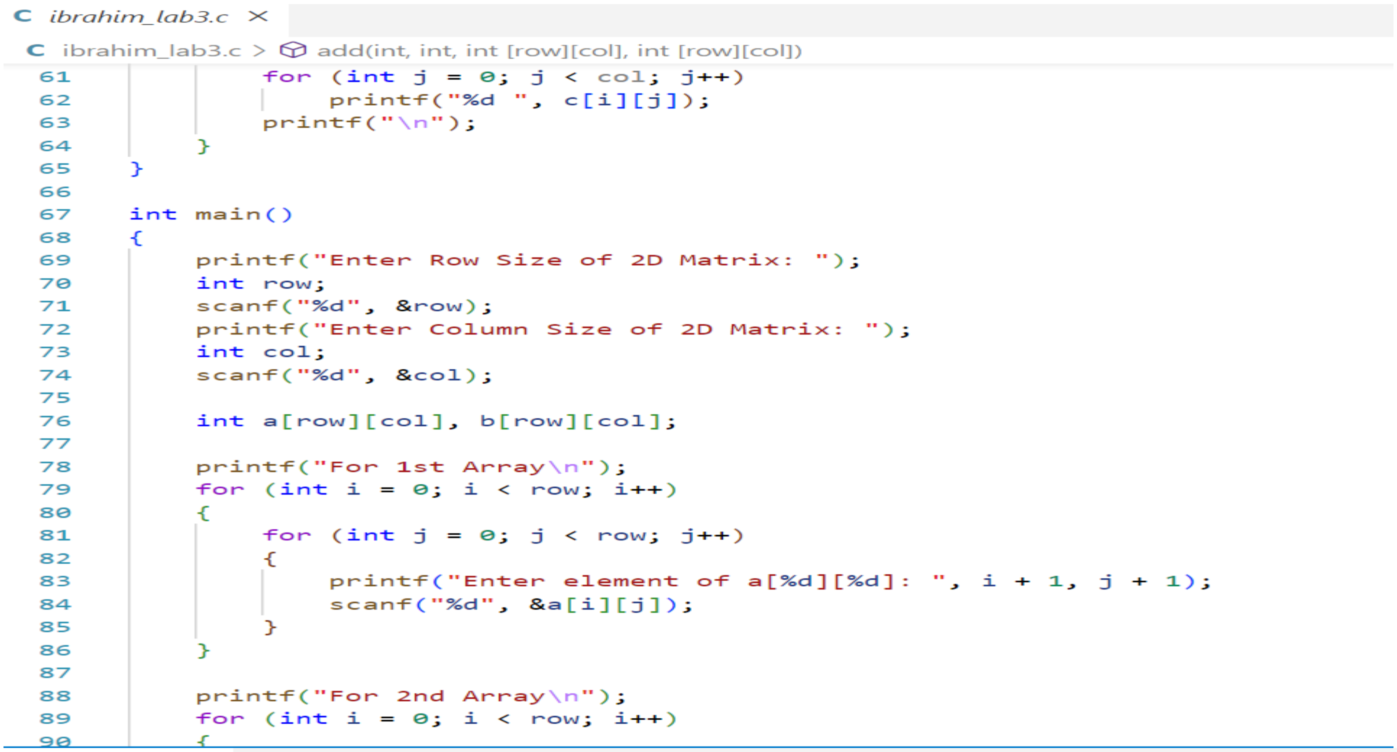
Program: 3


ibrahim_lab3.c
ibrahim_lab3.c > add(int, int, int [row][col], int [row][col])

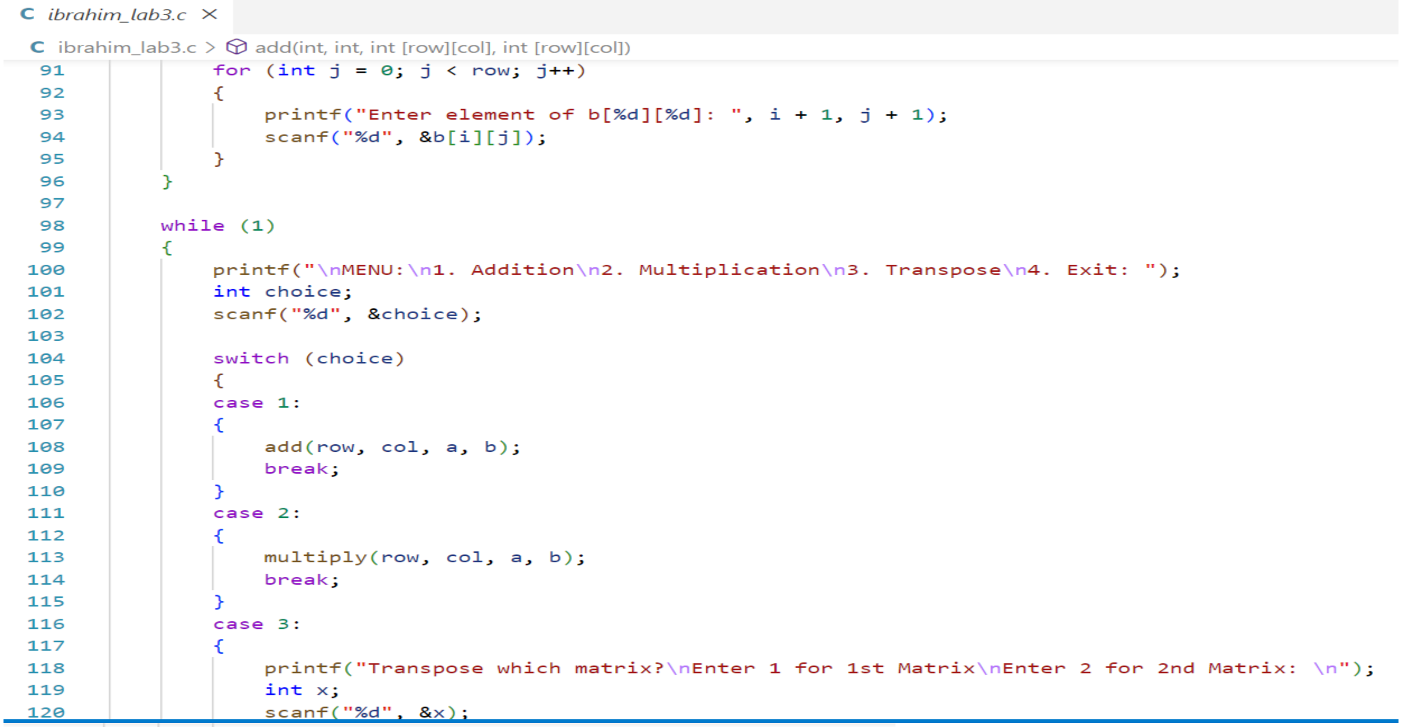
```c
// matrix

#include <stdio.h>

void add(int row, int col, int a[row][col], int b[row][col])
{
    int c[row][col];
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            c[i][j] = a[i][j] + b[i][j];
        }
    }
    printf("After addition, Result Matrix:\n");
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
            printf("%d ", c[i][j]);
        printf("\n");
    }
}

void multiply(int row, int col, int a[row][col], int b[row][col])
{
    int c[row][col];
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
```


ibrahim_lab3.c
ibrahim_lab3.c > add(int, int, int [row][col], int [row][col])

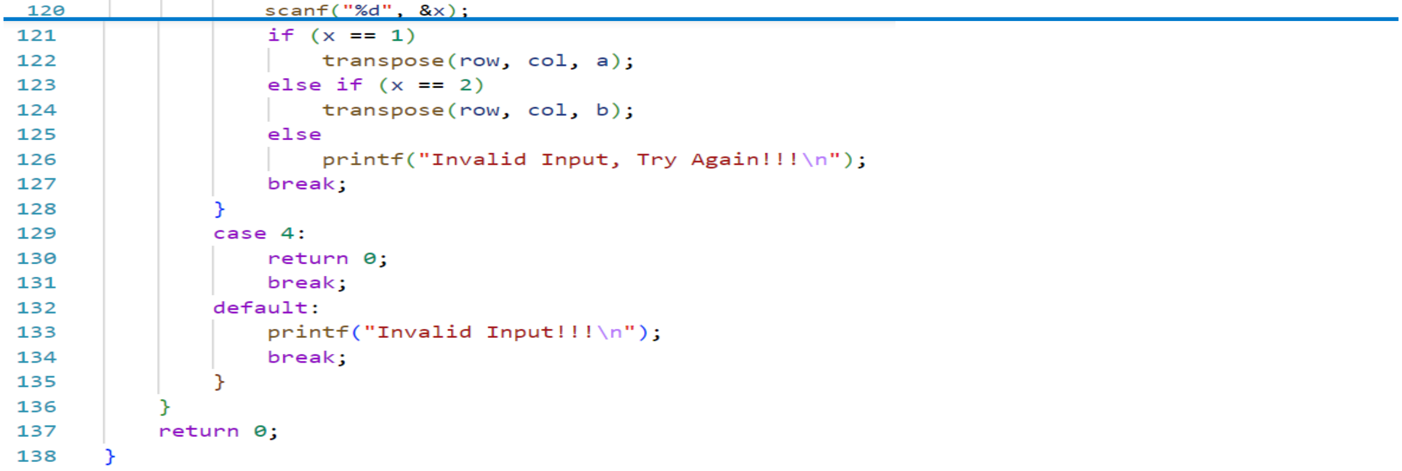
```c
            int temp = 0;
            for (int k = 0; k < row; k++)
            {
                temp = temp + (a[i][k] * b[k][j]);
                c[i][j] = temp;
            }
        }
    }
    printf("After multiplication, Result Matrix:\n");
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
            printf("%d ", c[i][j]);
        printf("\n");
    }
}

void transpose(int row, int col, int a[row][col])
{
    int c[row][col];
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            c[i][j] = a[j][i];
        }
    }
    printf("After transpose, Result Matrix:\n");
    for (int i = 0; i < row; i++)
    {
```


C ibrahim_lab3.c ×
C ibrahim_lab3.c > add(int, int, int [row][col], int [row][col])

```c
        for (int j = 0; j < col; j++)
            printf("%d ", c[i][j]);
        printf("\n");
    }
}

int main()
{
    printf("Enter Row Size of 2D Matrix: ");
    int row;
    scanf("%d", &row);
    printf("Enter Column Size of 2D Matrix: ");
    int col;
    scanf("%d", &col);

    int a[row][col], b[row][col];

    printf("For 1st Array\n");
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < row; j++)
        {
            printf("Enter element of a[%d][%d]: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }
    }

    printf("For 2nd Array\n");
    for (int i = 0; i < row; i++)
    {
```

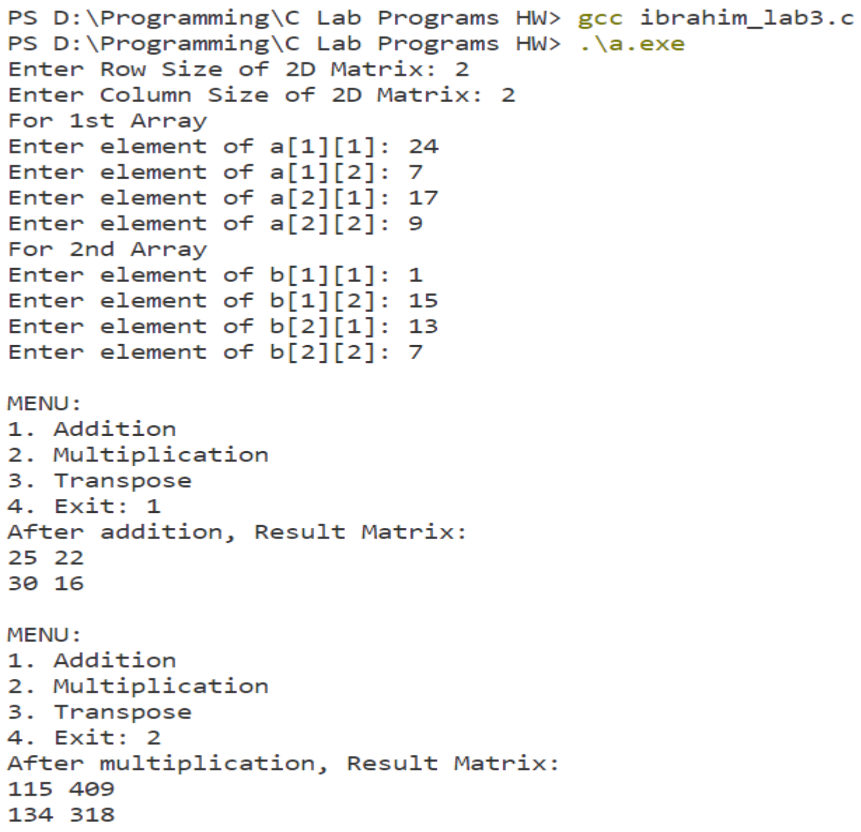

C ibrahim_lab3.c ×
C ibrahim_lab3.c > add(int, int, int [row][col], int [row][col])

```c
        for (int j = 0; j < row; j++)
        {
            printf("Enter element of b[%d][%d]: ", i + 1, j + 1);
            scanf("%d", &b[i][j]);
        }
    }

    while (1)
    {
        printf("\nMENU:\n1. Addition\n2. Multiplication\n3. Transpose\n4. Exit: ");
        int choice;
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
        {
            add(row, col, a, b);
            break;
        }
        case 2:
        {
            multiply(row, col, a, b);
            break;
        }
        case 3:
        {
            printf("Transpose which matrix?\nEnter 1 for 1st Matrix\nEnter 2 for 2nd Matrix: \n");
            int x;
            scanf("%d", &x);
```

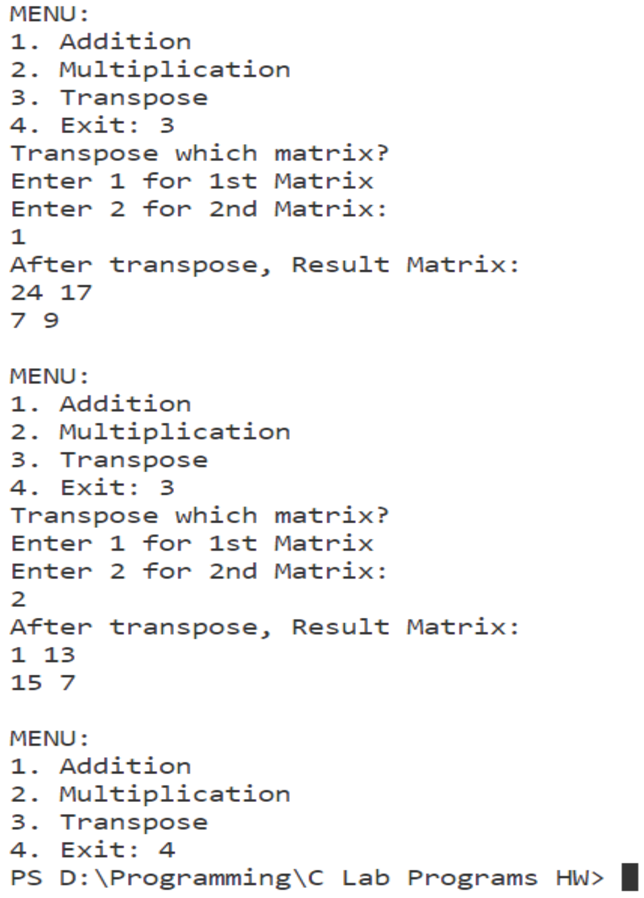


```c
            if (x == 1)
                transpose(row, col, a);
            else if (x == 2)
                transpose(row, col, b);
            else
                printf("Invalid Input, Try Again!!!\n");
            break;
        }
        case 4:
            return 0;
            break;
        default:
            printf("Invalid Input!!!\n");
            break;
        }
    }
    return 0;
}
```

Output: 3

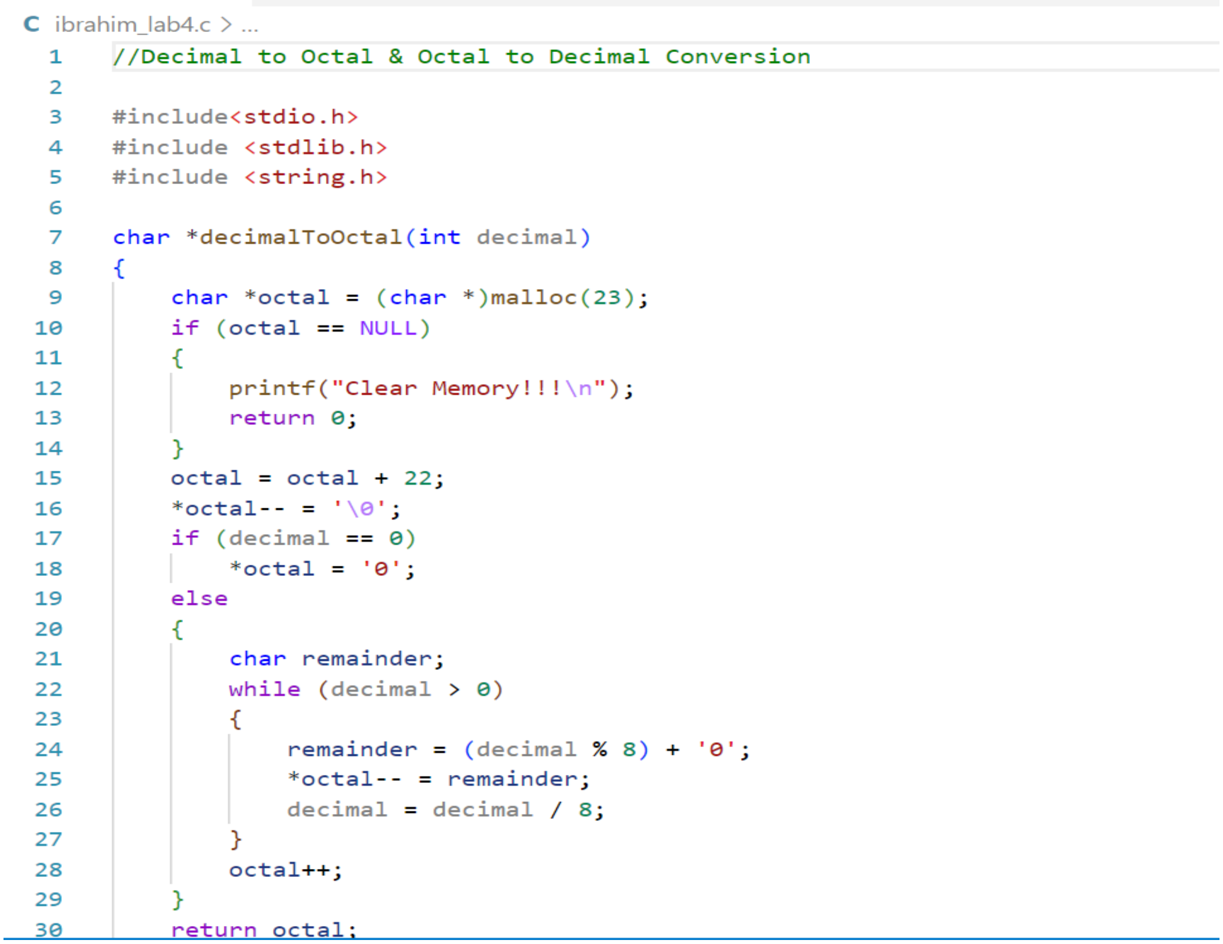


```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab3.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter Row Size of 2D Matrix: 2
Enter Column Size of 2D Matrix: 2
For 1st Array
Enter element of a[1][1]: 24
Enter element of a[1][2]: 7
Enter element of a[2][1]: 17
Enter element of a[2][2]: 9
For 2nd Array
Enter element of b[1][1]: 1
Enter element of b[1][2]: 15
Enter element of b[2][1]: 13
Enter element of b[2][2]: 7

MENU:
1. Addition
2. Multiplication
3. Transpose
4. Exit: 1
After addition, Result Matrix:
25 22
30 16

MENU:
1. Addition
2. Multiplication
3. Transpose
4. Exit: 2
After multiplication, Result Matrix:
115 409
134 318
```



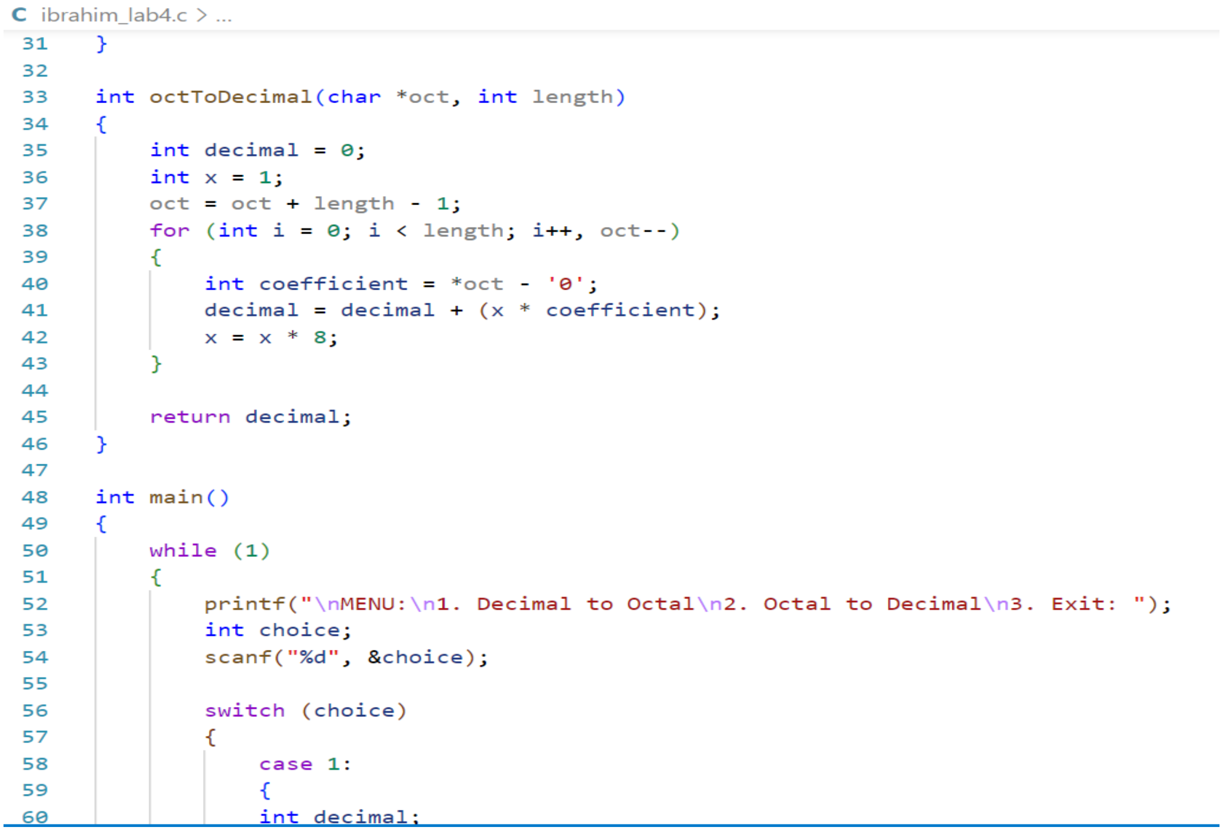
```
MENU:
1. Addition
2. Multiplication
3. Transpose
4. Exit: 3
Transpose which matrix?
Enter 1 for 1st Matrix
Enter 2 for 2nd Matrix:
1
After transpose, Result Matrix:
24 17
7 9

MENU:
1. Addition
2. Multiplication
3. Transpose
4. Exit: 3
Transpose which matrix?
Enter 1 for 1st Matrix
Enter 2 for 2nd Matrix:
2
After transpose, Result Matrix:
1 13
15 7

MENU:
1. Addition
2. Multiplication
3. Transpose
4. Exit: 4
PS D:\Programming\C Lab Programs HW>
```

Program: 4



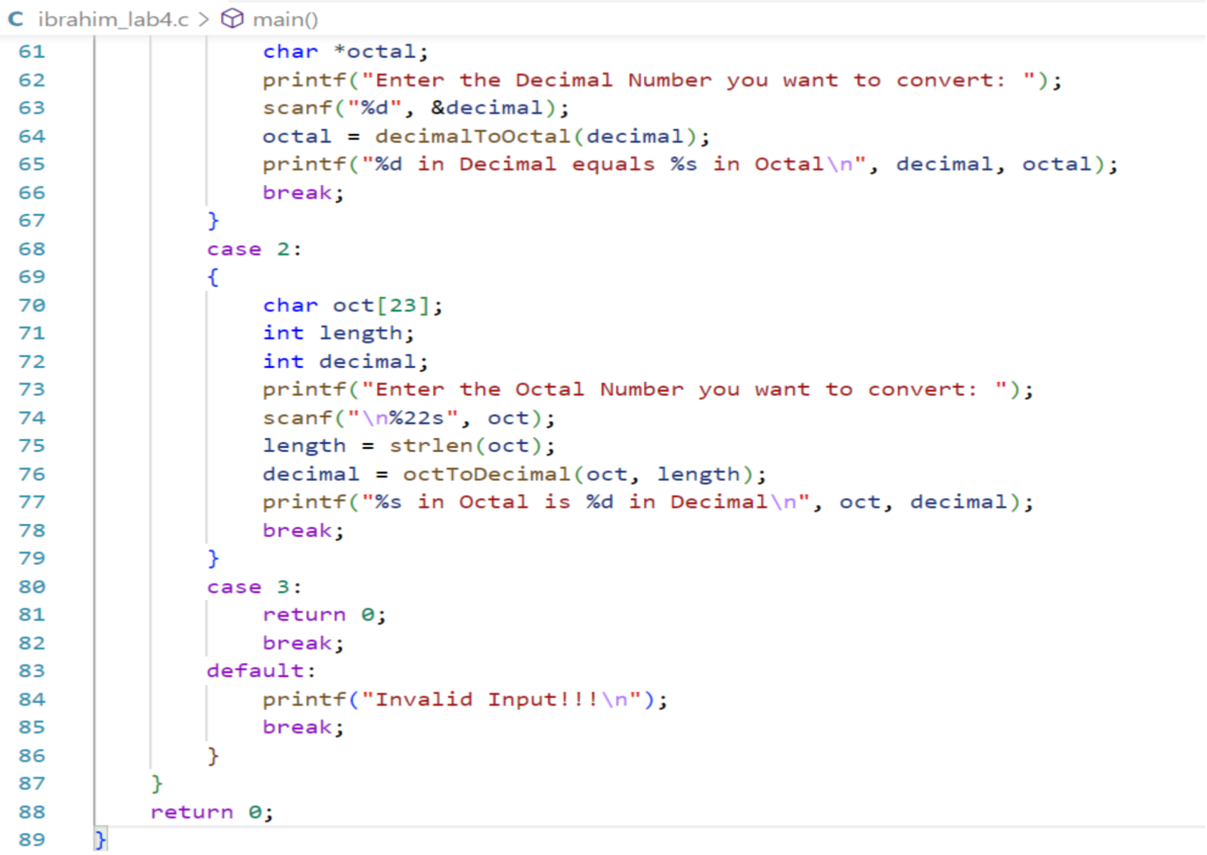
ibrahim_lab4.c > ...

```c
//Decimal to Octal & Octal to Decimal Conversion

#include<stdio.h>
#include <stdlib.h>
#include <string.h>

char *decimalToOctal(int decimal)
{
    char *octal = (char *)malloc(23);
    if (octal == NULL)
    {
        printf("Clear Memory!!!\n");
        return 0;
    }
    octal = octal + 22;
    *octal-- = '\0';
    if (decimal == 0)
        *octal = '0';
    else
    {
        char remainder;
        while (decimal > 0)
        {
            remainder = (decimal % 8) + '0';
            *octal-- = remainder;
            decimal = decimal / 8;
        }
        octal++;
    }
    return octal;
```



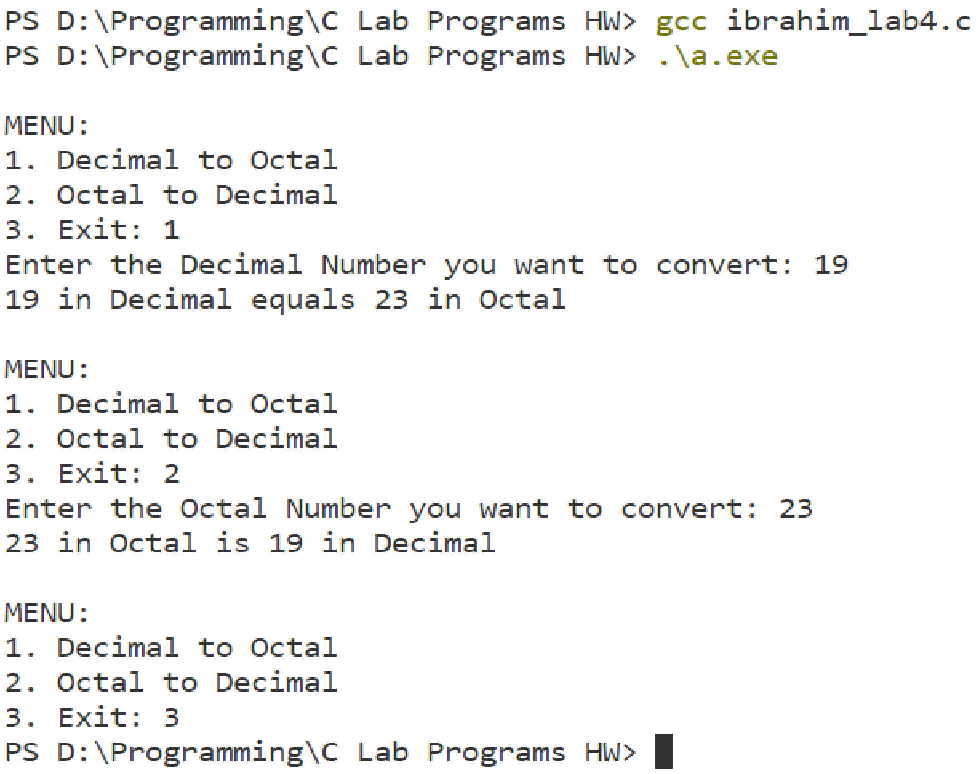
ibrahim_lab4.c > ...

```c
}

int octToDecimal(char *oct, int length)
{
    int decimal = 0;
    int x = 1;
    oct = oct + length - 1;
    for (int i = 0; i < length; i++, oct--)
    {
        int coefficient = *oct - '0';
        decimal = decimal + (x * coefficient);
        x = x * 8;
    }

    return decimal;
}

int main()
{
    while (1)
    {
        printf("\nMENU:\n1. Decimal to Octal\n2. Octal to Decimal\n3. Exit: ");
        int choice;
        scanf("%d", &choice);

        switch (choice)
        {
            case 1:
            {
            int decimal;
```



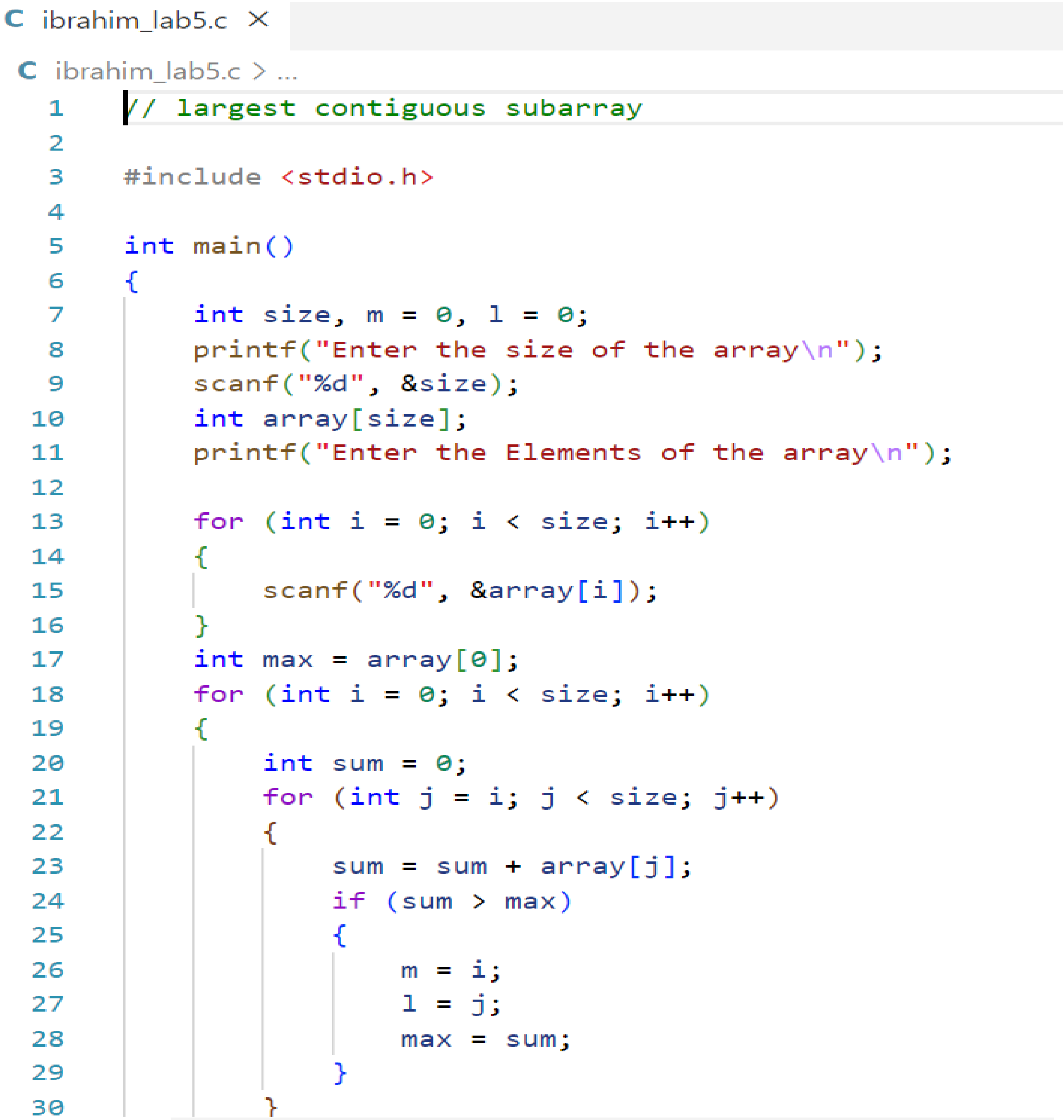
```
            char *octal;
            printf("Enter the Decimal Number you want to convert: ");
            scanf("%d", &decimal);
            octal = decimalToOctal(decimal);
            printf("%d in Decimal equals %s in Octal\n", decimal, octal);
            break;
        }
        case 2:
        {
            char oct[23];
            int length;
            int decimal;
            printf("Enter the Octal Number you want to convert: ");
            scanf("\n%22s", oct);
            length = strlen(oct);
            decimal = octToDecimal(oct, length);
            printf("%s in Octal is %d in Decimal\n", oct, decimal);
            break;
        }
        case 3:
            return 0;
            break;
        default:
            printf("Invalid Input!!!\n");
            break;
        }
    }
    return 0;
}
```

Output: 4

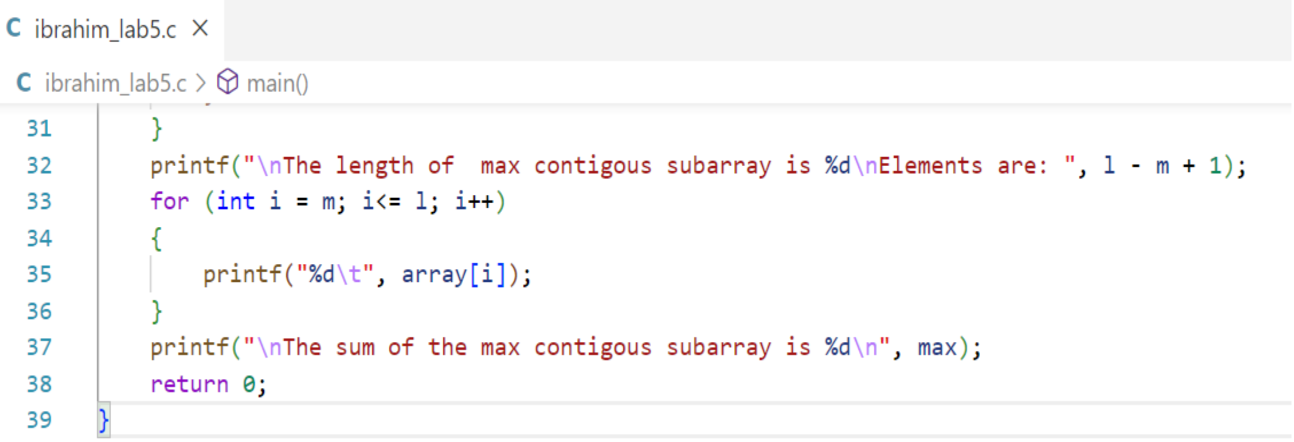


```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab4.c
PS D:\Programming\C Lab Programs HW> .\a.exe

MENU:
1. Decimal to Octal
2. Octal to Decimal
3. Exit: 1
Enter the Decimal Number you want to convert: 19
19 in Decimal equals 23 in Octal

MENU:
1. Decimal to Octal
2. Octal to Decimal
3. Exit: 2
Enter the Octal Number you want to convert: 23
23 in Octal is 19 in Decimal

MENU:
1. Decimal to Octal
2. Octal to Decimal
3. Exit: 3
PS D:\Programming\C Lab Programs HW>
```

Program: 5

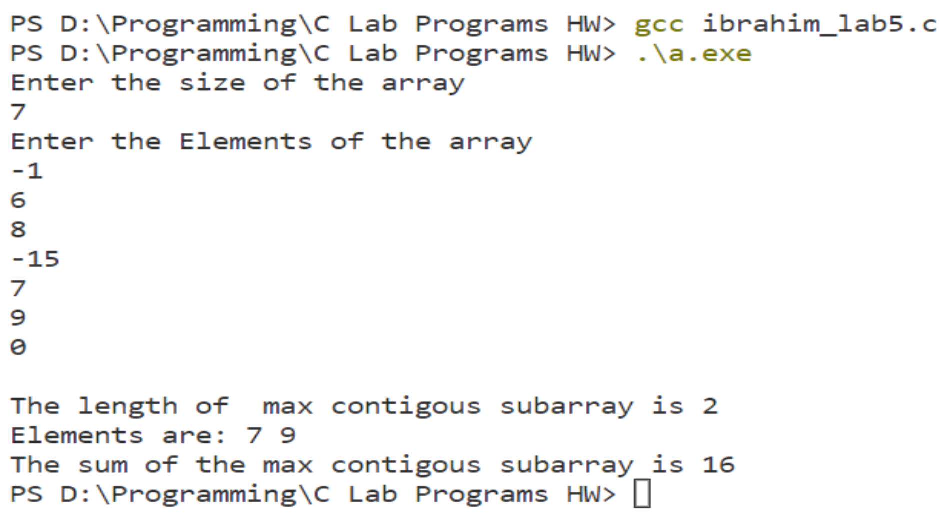


```c
// largest contiguous subarray

#include <stdio.h>

int main()
{
    int size, m = 0, l = 0;
    printf("Enter the size of the array\n");
    scanf("%d", &size);
    int array[size];
    printf("Enter the Elements of the array\n");

    for (int i = 0; i < size; i++)
    {
        scanf("%d", &array[i]);
    }
    int max = array[0];
    for (int i = 0; i < size; i++)
    {
        int sum = 0;
        for (int j = i; j < size; j++)
        {
            sum = sum + array[j];
            if (sum > max)
            {
                m = i;
                l = j;
                max = sum;
            }
        }
```



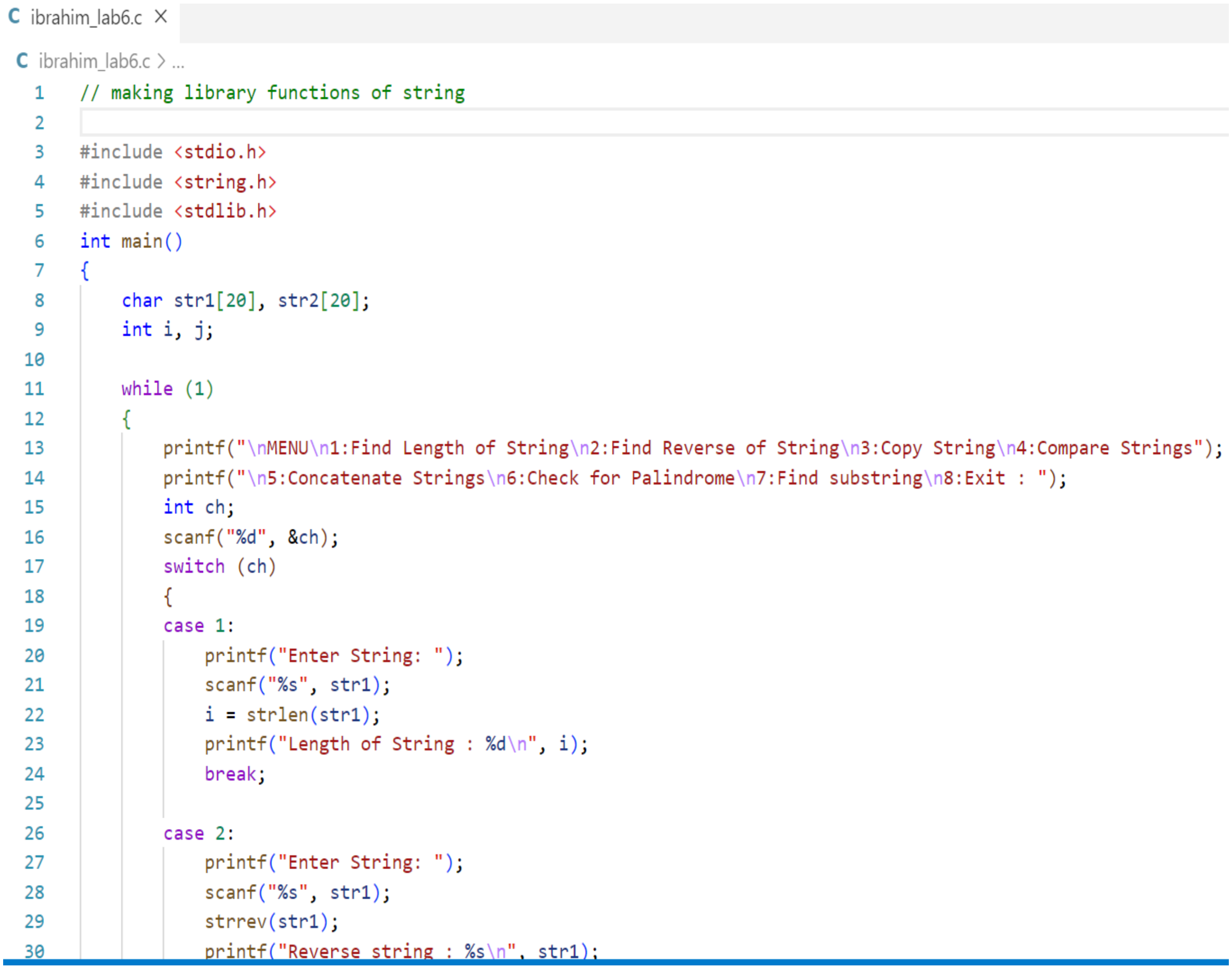
```c
    }
    printf("\nThe length of  max contigous subarray is %d\nElements are: ", l - m + 1);
    for (int i = m; i<= l; i++)
    {
        printf("%d\t", array[i]);
    }
    printf("\nThe sum of the max contigous subarray is %d\n", max);
    return 0;
}
```

Output: 5

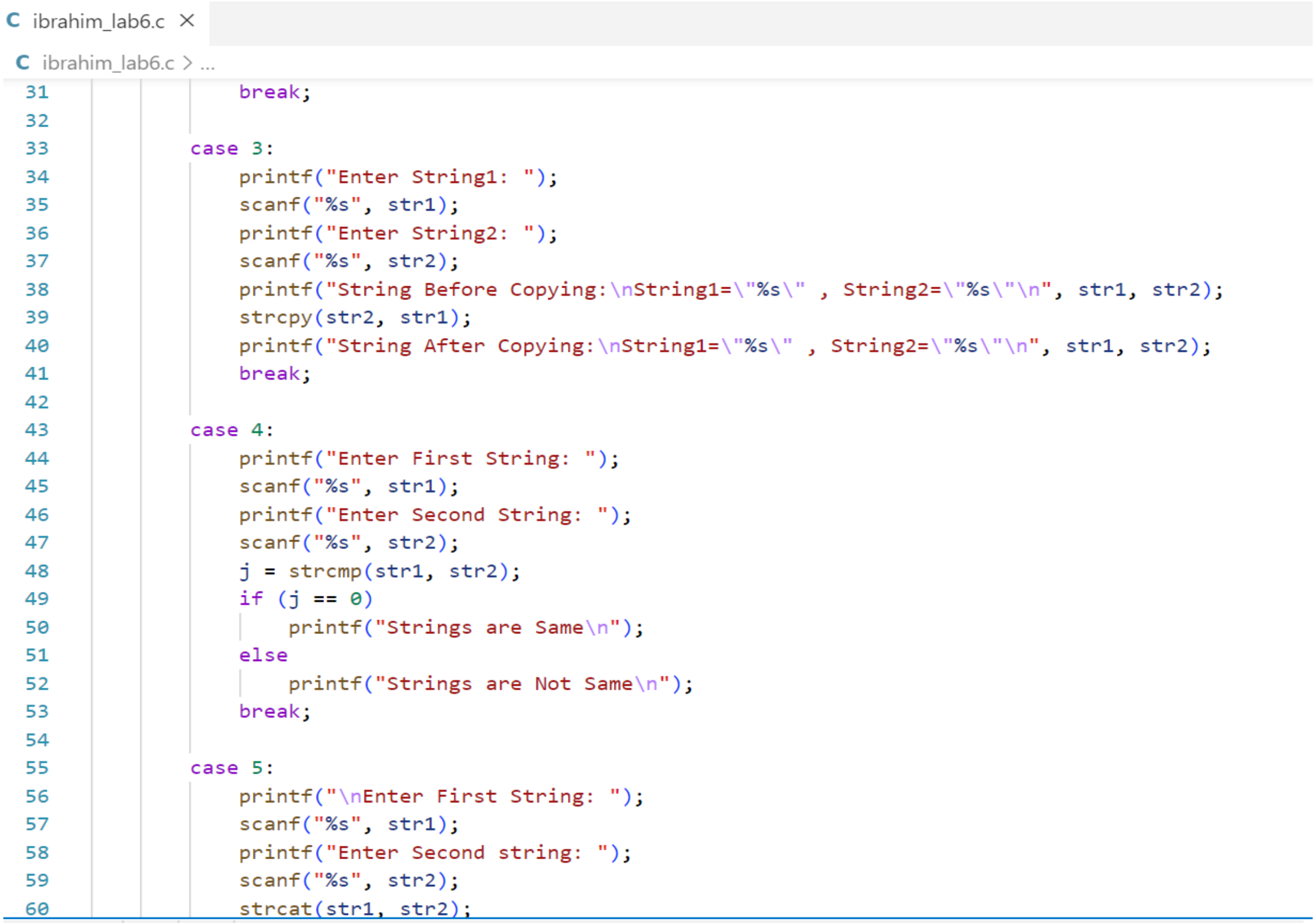


```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab5.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter the size of the array
7
Enter the Elements of the array
-1
6
8
-15
7
9
0

The length of  max contigous subarray is 2
Elements are: 7 9
The sum of the max contigous subarray is 16
PS D:\Programming\C Lab Programs HW> 
```

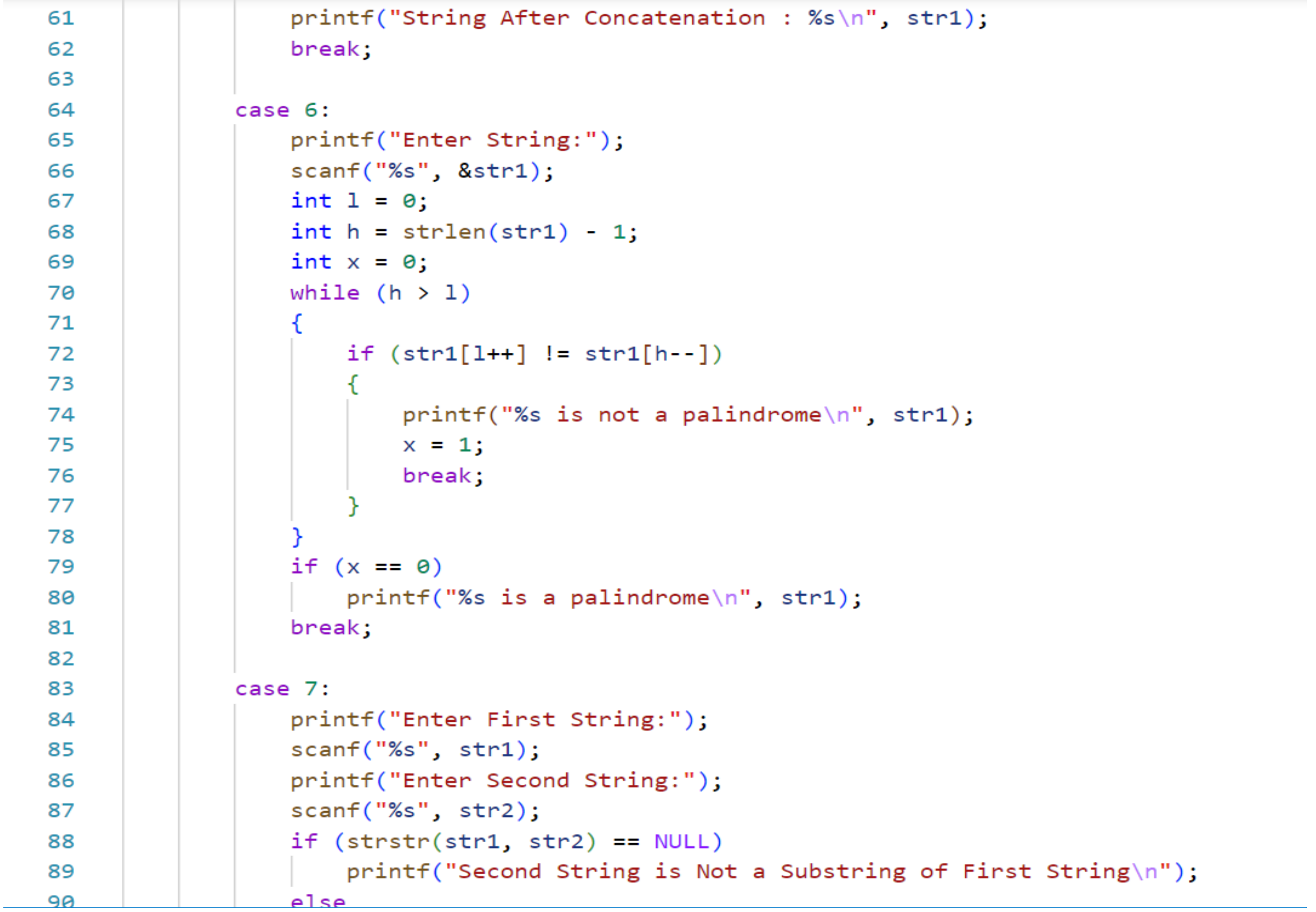
Program: 6



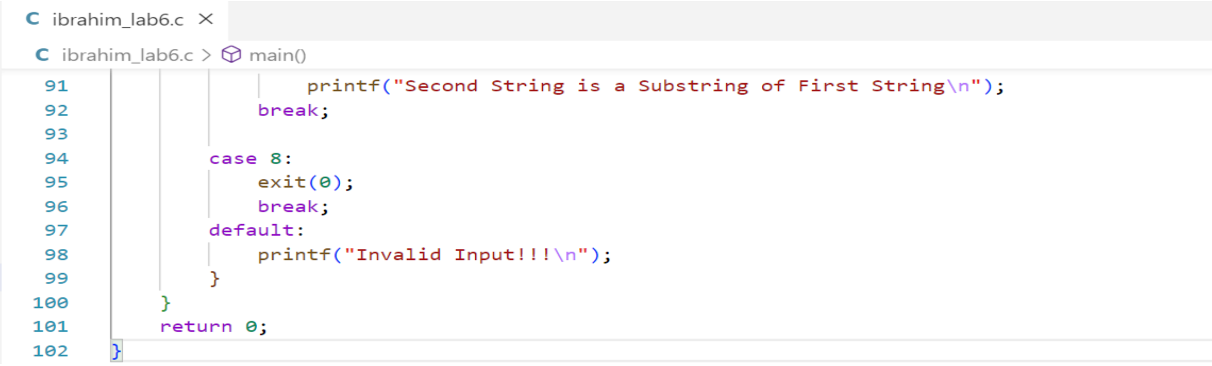
```c
// making library functions of string

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int main()
{
    char str1[20], str2[20];
    int i, j;

    while (1)
    {
        printf("\nMENU\n1:Find Length of String\n2:Find Reverse of String\n3:Copy String\n4:Compare Strings");
        printf("\n5:Concatenate Strings\n6:Check for Palindrome\n7:Find substring\n8:Exit : ");
        int ch;
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:
            printf("Enter String: ");
            scanf("%s", str1);
            i = strlen(str1);
            printf("Length of String : %d\n", i);
            break;

        case 2:
            printf("Enter String: ");
            scanf("%s", str1);
            strrev(str1);
            printf("Reverse string : %s\n", str1);
```



```
                break;

            case 3:
                printf("Enter String1: ");
                scanf("%s", str1);
                printf("Enter String2: ");
                scanf("%s", str2);
                printf("String Before Copying:\nString1=\"%s\" , String2=\"%s\"\n", str1, str2);
                strcpy(str2, str1);
                printf("String After Copying:\nString1=\"%s\" , String2=\"%s\"\n", str1, str2);
                break;

            case 4:
                printf("Enter First String: ");
                scanf("%s", str1);
                printf("Enter Second String: ");
                scanf("%s", str2);
                j = strcmp(str1, str2);
                if (j == 0)
                    printf("Strings are Same\n");
                else
                    printf("Strings are Not Same\n");
                break;

            case 5:
                printf("\nEnter First String: ");
                scanf("%s", str1);
                printf("Enter Second string: ");
                scanf("%s", str2);
                strcat(str1, str2);
```

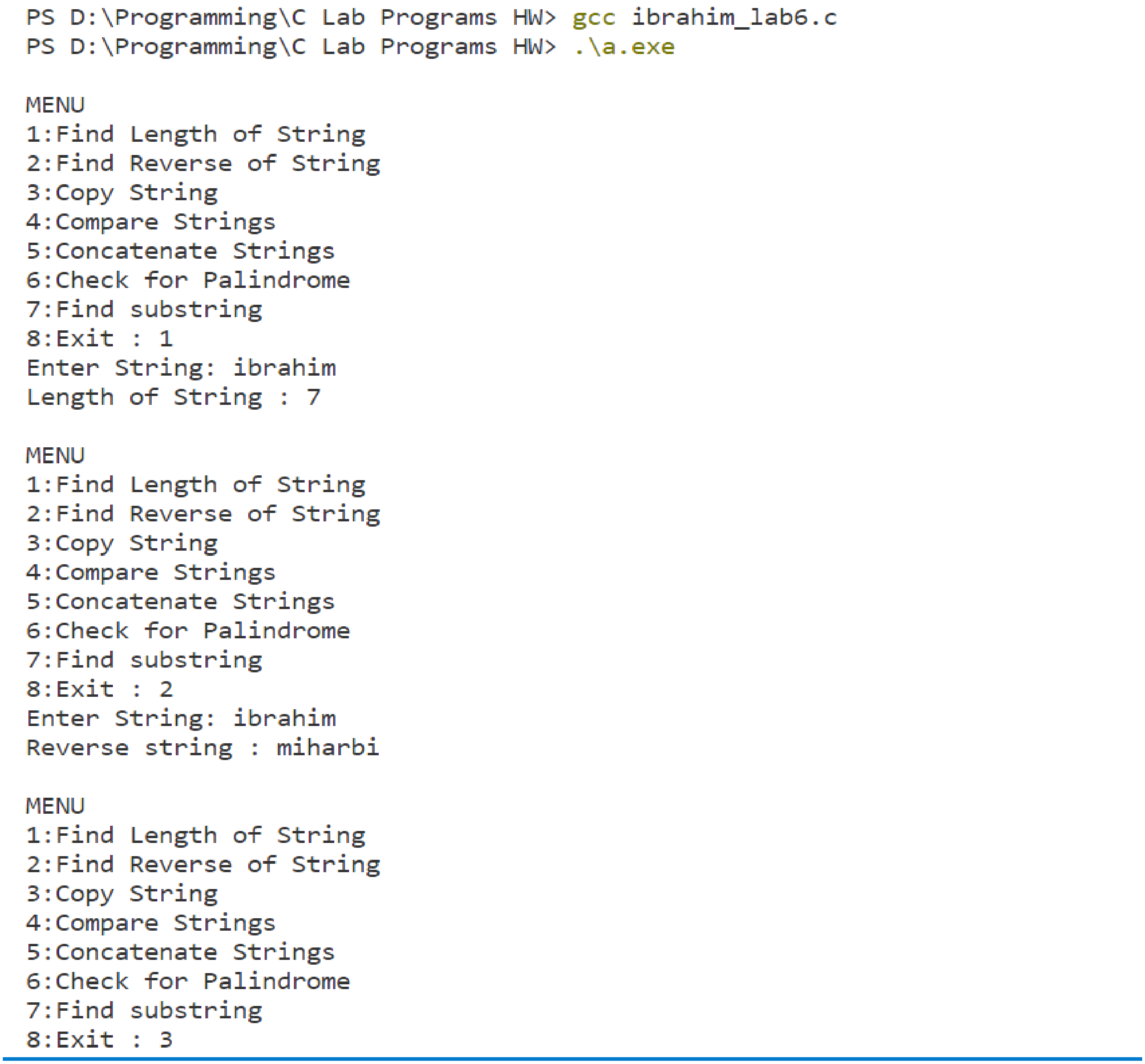


```
                printf("String After Concatenation : %s\n", str1);
                break;

            case 6:
                printf("Enter String:");
                scanf("%s", &str1);
                int l = 0;
                int h = strlen(str1) - 1;
                int x = 0;
                while (h > l)
                {
                    if (str1[l++] != str1[h--])
                    {
                        printf("%s is not a palindrome\n", str1);
                        x = 1;
                        break;
                    }
                }
                if (x == 0)
                    printf("%s is a palindrome\n", str1);
                break;

            case 7:
                printf("Enter First String:");
                scanf("%s", str1);
                printf("Enter Second String:");
                scanf("%s", str2);
                if (strstr(str1, str2) == NULL)
                    printf("Second String is Not a Substring of First String\n");
                else
```

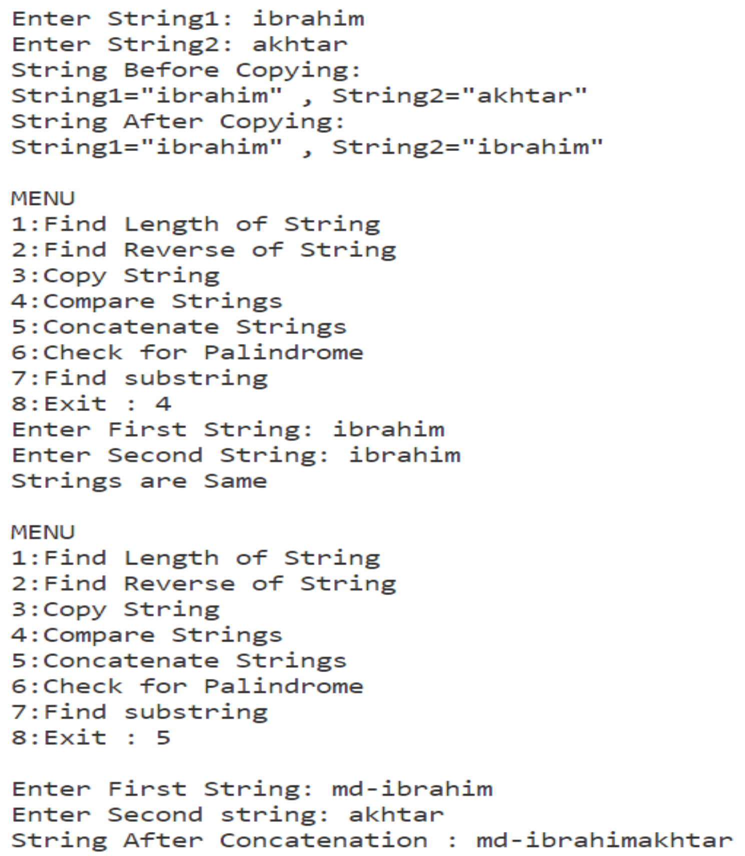


```
                printf("Second String is a Substring of First String\n");
            break;

        case 8:
            exit(0);
            break;
        default:
            printf("Invalid Input!!!\n");
        }
    }
    return 0;
}
```

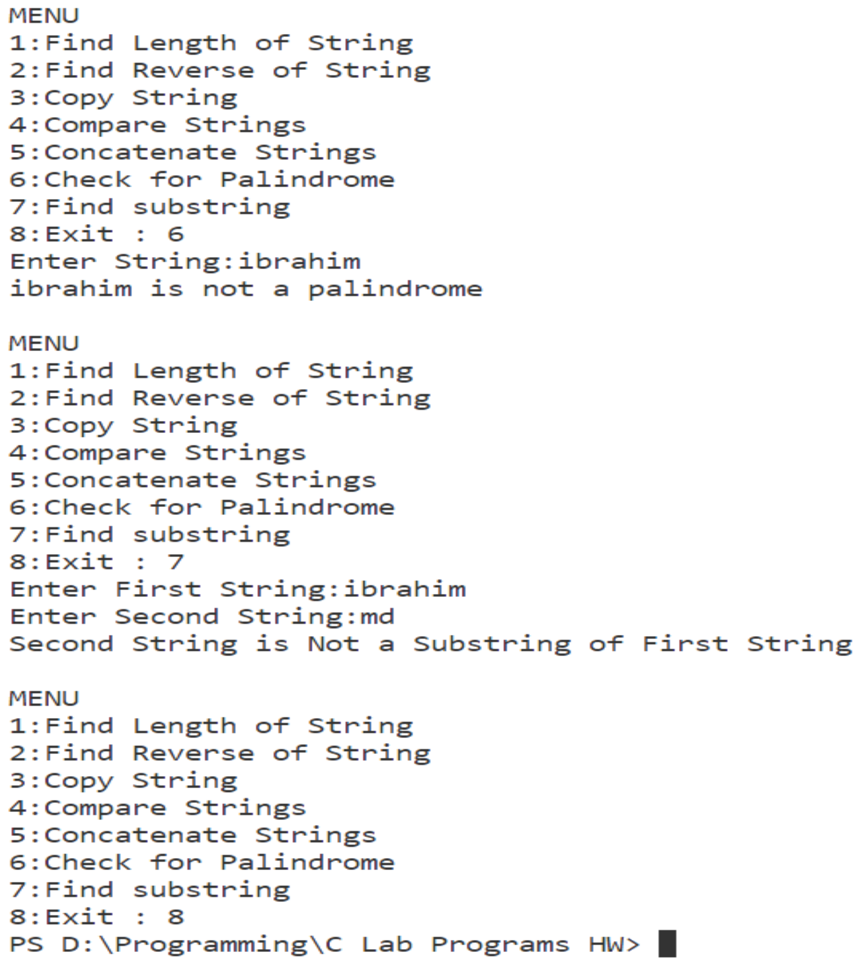
Output: 6



```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab6.c
PS D:\Programming\C Lab Programs HW> .\a.exe

MENU
1:Find Length of String
2:Find Reverse of String
3:Copy String
4:Compare Strings
5:Concatenate Strings
6:Check for Palindrome
7:Find substring
8:Exit : 1
Enter String: ibrahim
Length of String : 7

MENU
1:Find Length of String
2:Find Reverse of String
3:Copy String
4:Compare Strings
5:Concatenate Strings
6:Check for Palindrome
7:Find substring
8:Exit : 2
Enter String: ibrahim
Reverse string : miharbi

MENU
1:Find Length of String
2:Find Reverse of String
3:Copy String
4:Compare Strings
5:Concatenate Strings
6:Check for Palindrome
7:Find substring
8:Exit : 3
```

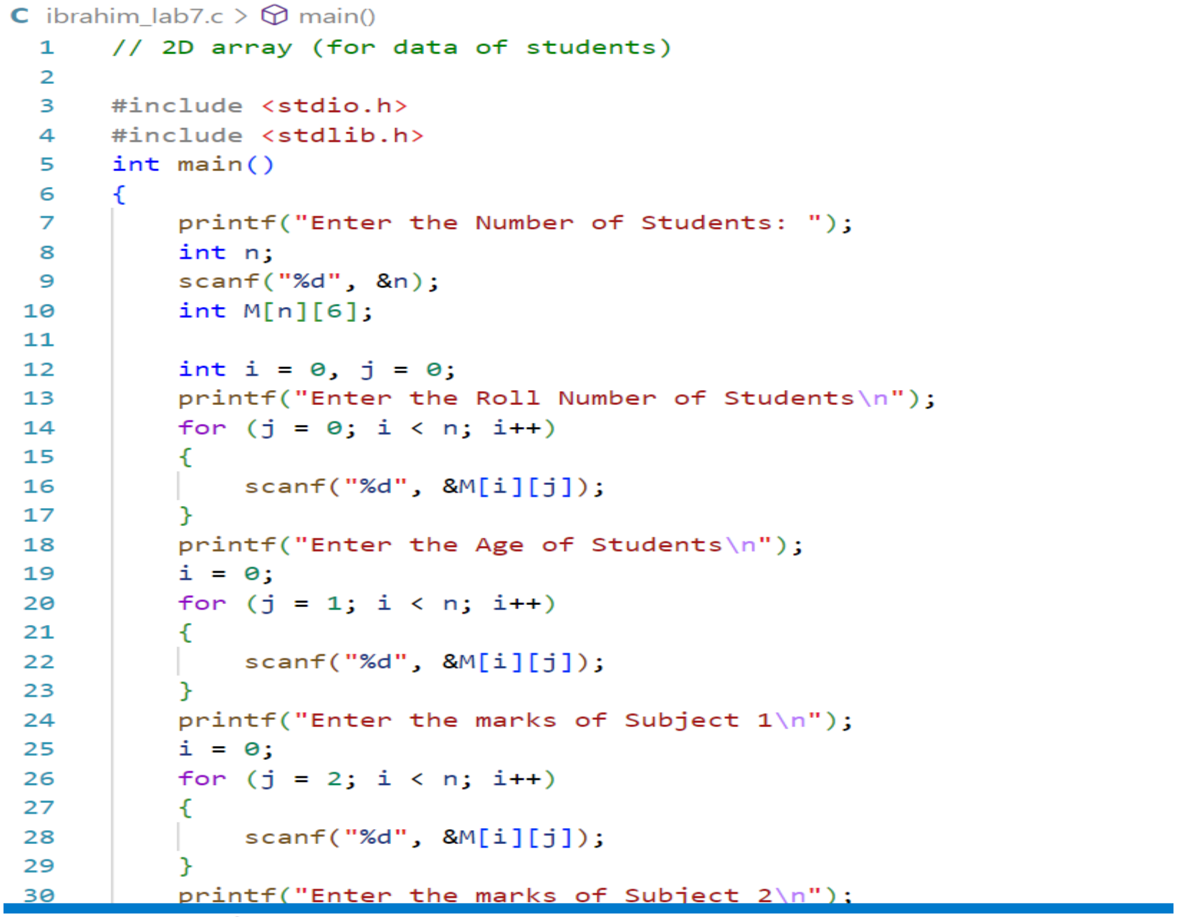


```
Enter String1: ibrahim
Enter String2: akhtar
String Before Copying:
String1="ibrahim" , String2="akhtar"
String After Copying:
String1="ibrahim" , String2="ibrahim"

MENU
1:Find Length of String
2:Find Reverse of String
3:Copy String
4:Compare Strings
5:Concatenate Strings
6:Check for Palindrome
7:Find substring
8:Exit : 4
Enter First String: ibrahim
Enter Second String: ibrahim
Strings are Same

MENU
1:Find Length of String
2:Find Reverse of String
3:Copy String
4:Compare Strings
5:Concatenate Strings
6:Check for Palindrome
7:Find substring
8:Exit : 5

Enter First String: md-ibrahim
Enter Second string: akhtar
String After Concatenation : md-ibrahimakhtar
```

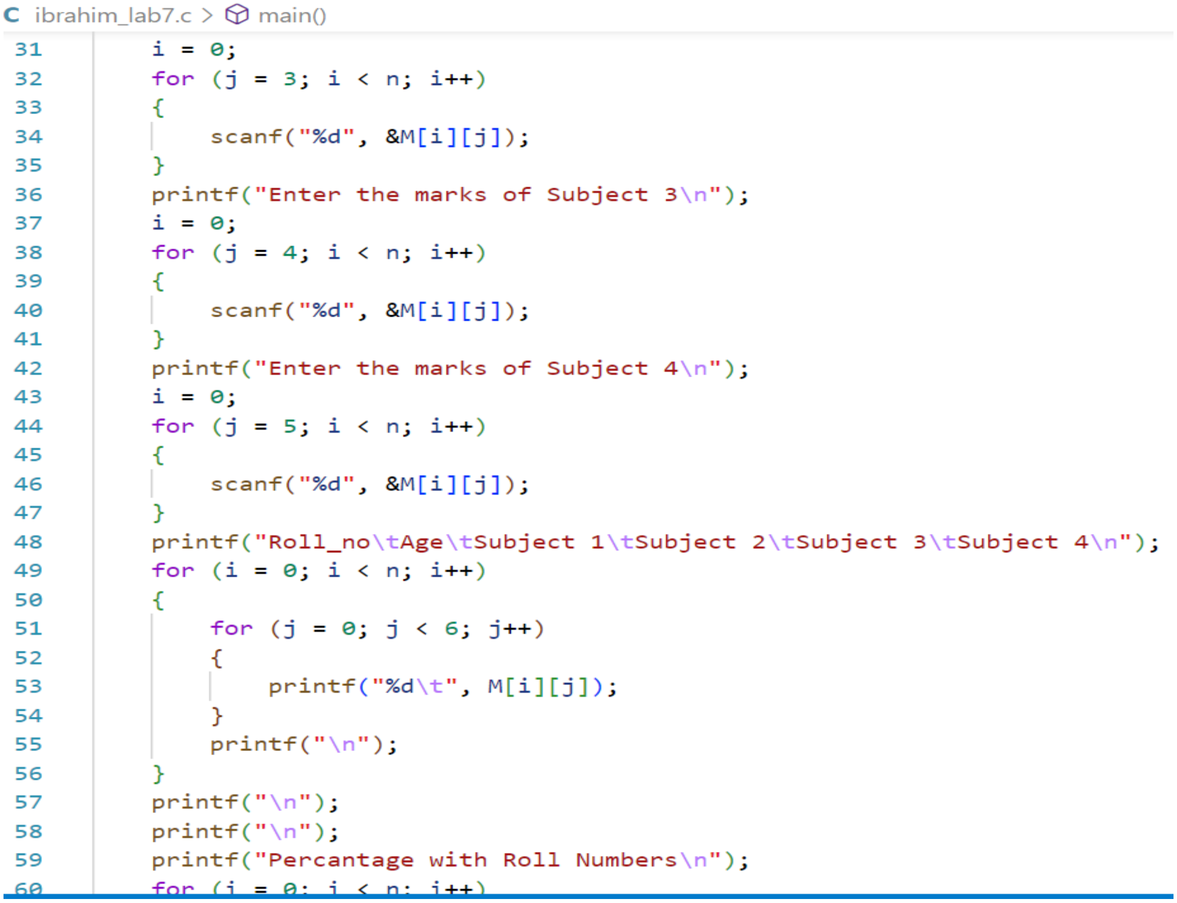


```
MENU
1:Find Length of String
2:Find Reverse of String
3:Copy String
4:Compare Strings
5:Concatenate Strings
6:Check for Palindrome
7:Find substring
8:Exit : 6
Enter String:ibrahim
ibrahim is not a palindrome

MENU
1:Find Length of String
2:Find Reverse of String
3:Copy String
4:Compare Strings
5:Concatenate Strings
6:Check for Palindrome
7:Find substring
8:Exit : 7
Enter First String:ibrahim
Enter Second String:md
Second String is Not a Substring of First String

MENU
1:Find Length of String
2:Find Reverse of String
3:Copy String
4:Compare Strings
5:Concatenate Strings
6:Check for Palindrome
7:Find substring
8:Exit : 8
PS D:\Programming\C Lab Programs HW> █
```

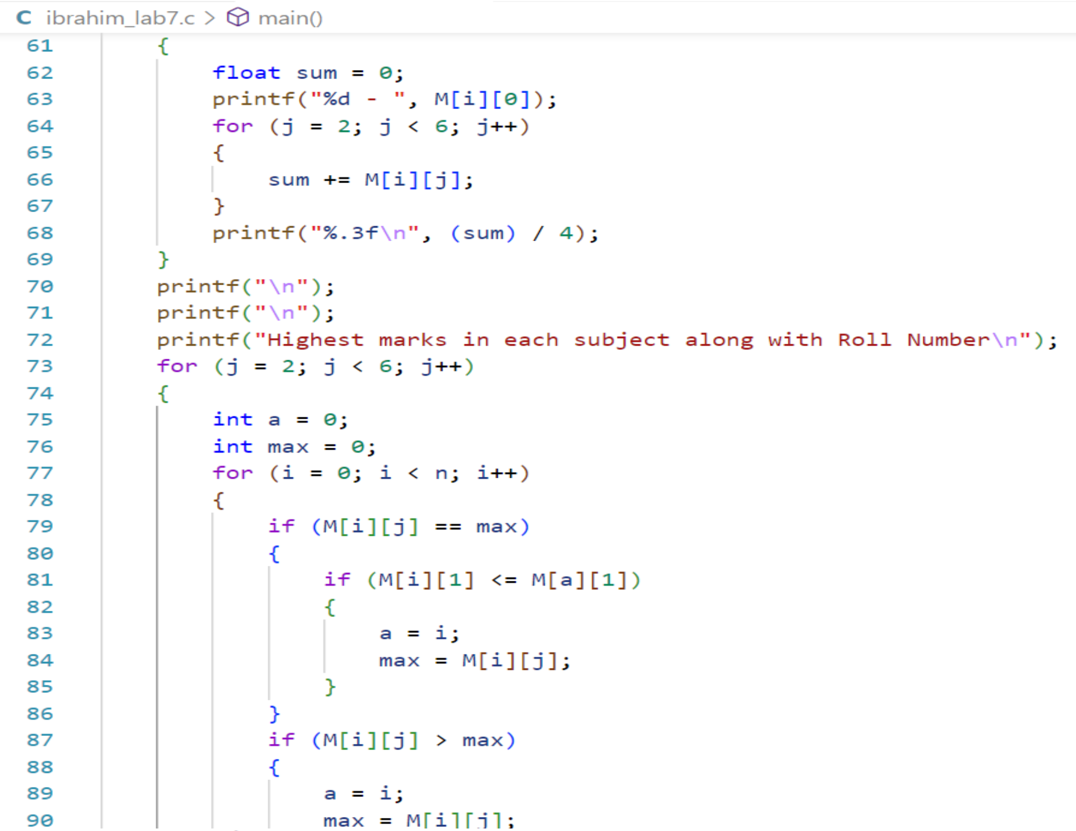
Program: 7



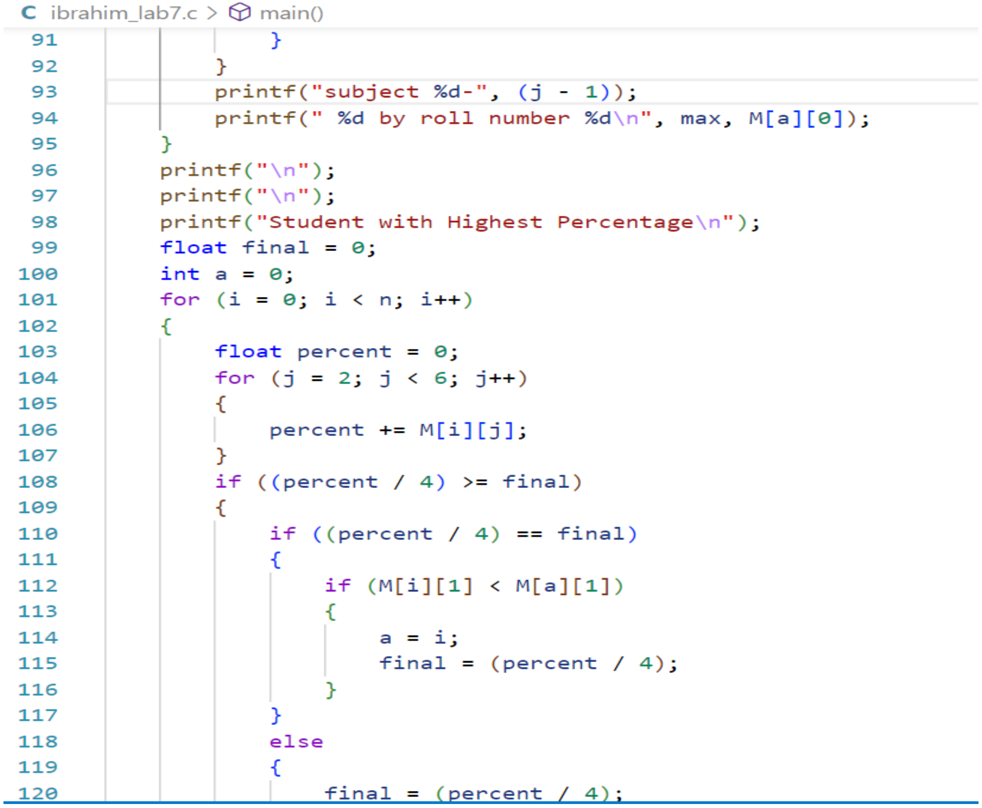
```c
// 2D array (for data of students)

#include <stdio.h>
#include <stdlib.h>
int main()
{
    printf("Enter the Number of Students: ");
    int n;
    scanf("%d", &n);
    int M[n][6];

    int i = 0, j = 0;
    printf("Enter the Roll Number of Students\n");
    for (j = 0; i < n; i++)
    {
        scanf("%d", &M[i][j]);
    }
    printf("Enter the Age of Students\n");
    i = 0;
    for (j = 1; i < n; i++)
    {
        scanf("%d", &M[i][j]);
    }
    printf("Enter the marks of Subject 1\n");
    i = 0;
    for (j = 2; i < n; i++)
    {
        scanf("%d", &M[i][j]);
    }
    printf("Enter the marks of Subject 2\n");
```



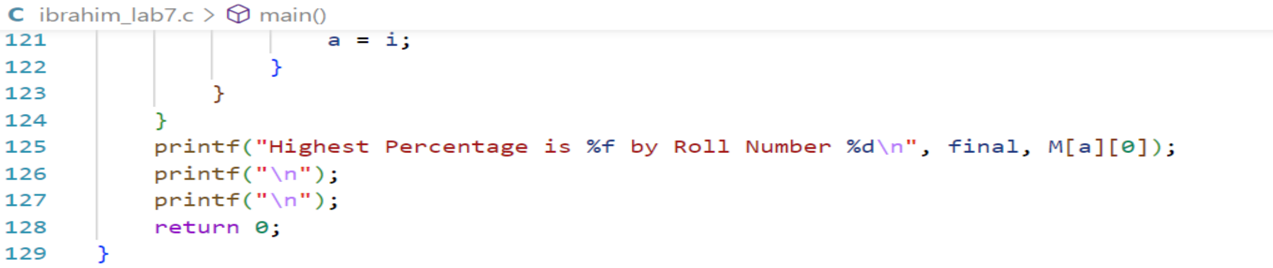
```c
    i = 0;
    for (j = 3; i < n; i++)
    {
        scanf("%d", &M[i][j]);
    }
    printf("Enter the marks of Subject 3\n");
    i = 0;
    for (j = 4; i < n; i++)
    {
        scanf("%d", &M[i][j]);
    }
    printf("Enter the marks of Subject 4\n");
    i = 0;
    for (j = 5; i < n; i++)
    {
        scanf("%d", &M[i][j]);
    }
    printf("Roll_no\tAge\tSubject 1\tSubject 2\tSubject 3\tSubject 4\n");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < 6; j++)
        {
            printf("%d\t", M[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    printf("\n");
    printf("Percantage with Roll Numbers\n");
    for (i = 0; i < n; i++)
```



```c
    {
        float sum = 0;
        printf("%d - ", M[i][0]);
        for (j = 2; j < 6; j++)
        {
            sum += M[i][j];
        }
        printf("%.3f\n", (sum) / 4);
    }
    printf("\n");
    printf("\n");
    printf("Highest marks in each subject along with Roll Number\n");
    for (j = 2; j < 6; j++)
    {
        int a = 0;
        int max = 0;
        for (i = 0; i < n; i++)
        {
            if (M[i][j] == max)
            {
                if (M[i][1] <= M[a][1])
                {
                    a = i;
                    max = M[i][j];
                }
            }
            if (M[i][j] > max)
            {
                a = i;
                max = M[i][j];
```

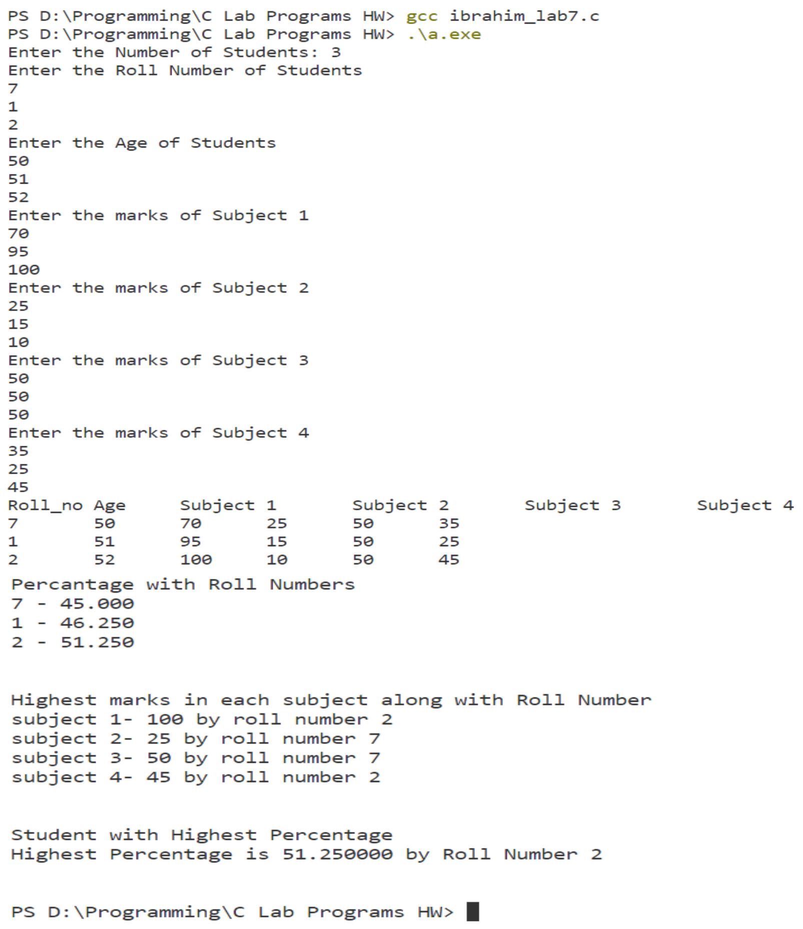


```c
            }
        }
        printf("subject %d-", (j - 1));
        printf(" %d by roll number %d\n", max, M[a][0]);
    }
    printf("\n");
    printf("\n");
    printf("Student with Highest Percentage\n");
    float final = 0;
    int a = 0;
    for (i = 0; i < n; i++)
    {
        float percent = 0;
        for (j = 2; j < 6; j++)
        {
            percent += M[i][j];
        }
        if ((percent / 4) >= final)
        {
            if ((percent / 4) == final)
            {
                if (M[i][1] < M[a][1])
                {
                    a = i;
                    final = (percent / 4);
                }
            }
            else
            {
                final = (percent / 4);
```



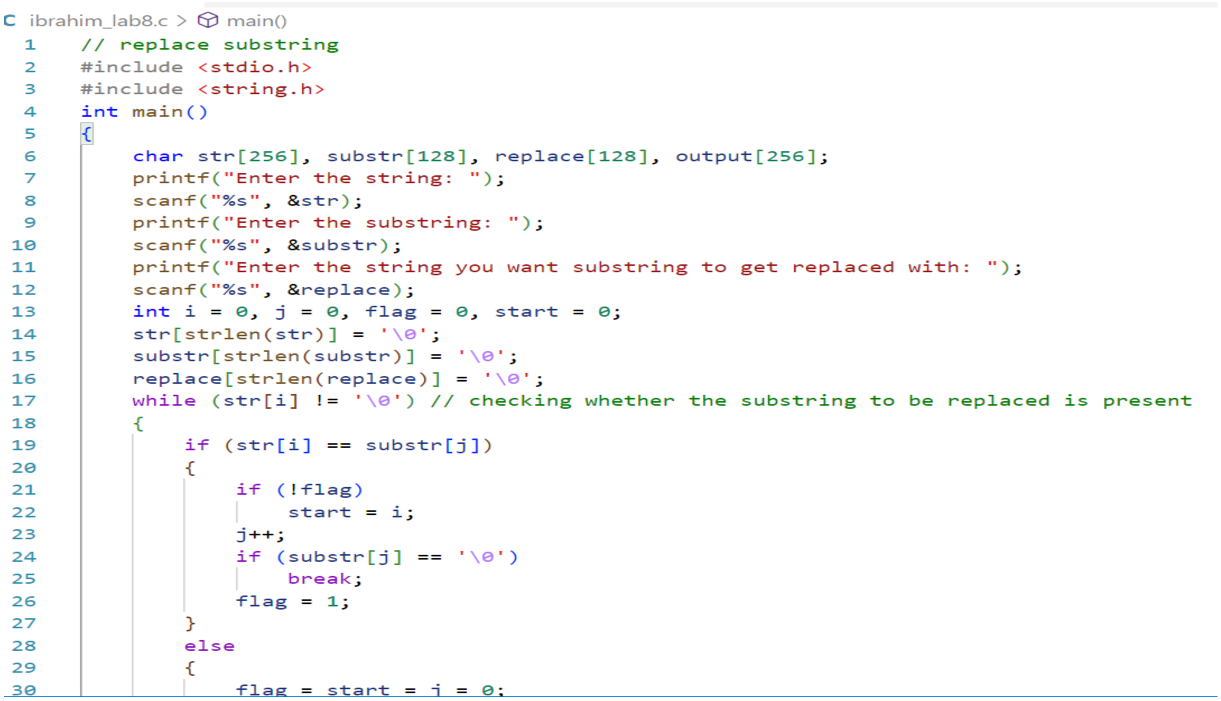
```c
                a = i;
            }
        }
    }
    printf("Highest Percentage is %f by Roll Number %d\n", final, M[a][0]);
    printf("\n");
    printf("\n");
    return 0;
}
```

Output: 7

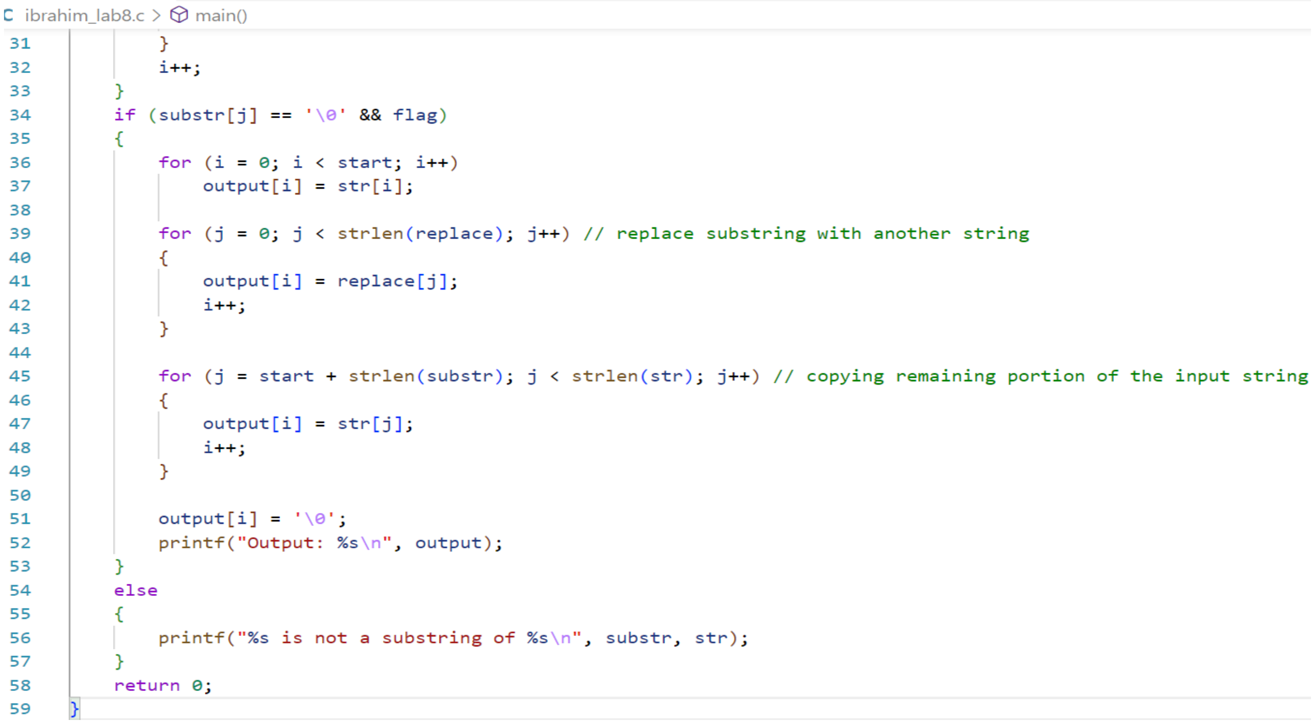


```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab7.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter the Number of Students: 3
Enter the Roll Number of Students
7
1
2
Enter the Age of Students
50
51
52
Enter the marks of Subject 1
70
95
100
Enter the marks of Subject 2
25
15
10
Enter the marks of Subject 3
50
50
50
Enter the marks of Subject 4
35
25
45
Roll_no Age     Subject 1       Subject 2       Subject 3       Subject 4
7       50      70      25      50      35
1       51      95      15      50      25
2       52      100     10      50      45
Percantage with Roll Numbers
7 - 45.000
1 - 46.250
2 - 51.250


Highest marks in each subject along with Roll Number
subject 1- 100 by roll number 2
subject 2- 25 by roll number 7
subject 3- 50 by roll number 7
subject 4- 45 by roll number 2


Student with Highest Percentage
Highest Percentage is 51.250000 by Roll Number 2

PS D:\Programming\C Lab Programs HW> █
```

Program: 8

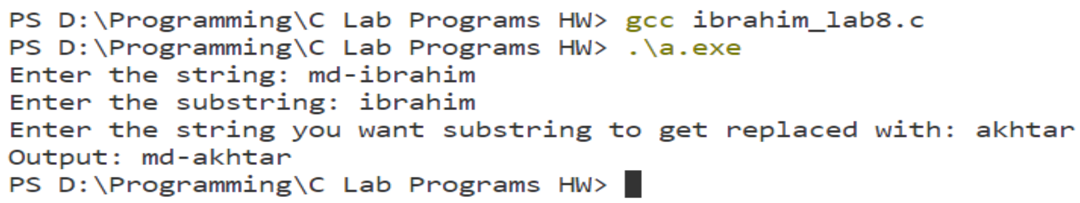


```c
// replace substring
#include <stdio.h>
#include <string.h>
int main()
{
    char str[256], substr[128], replace[128], output[256];
    printf("Enter the string: ");
    scanf("%s", &str);
    printf("Enter the substring: ");
    scanf("%s", &substr);
    printf("Enter the string you want substring to get replaced with: ");
    scanf("%s", &replace);
    int i = 0, j = 0, flag = 0, start = 0;
    str[strlen(str)] = '\0';
    substr[strlen(substr)] = '\0';
    replace[strlen(replace)] = '\0';
    while (str[i] != '\0') // checking whether the substring to be replaced is present
    {
        if (str[i] == substr[j])
        {
            if (!flag)
                start = i;
            j++;
            if (substr[j] == '\0')
                break;
            flag = 1;
        }
        else
        {
            flag = start = j = 0;
```



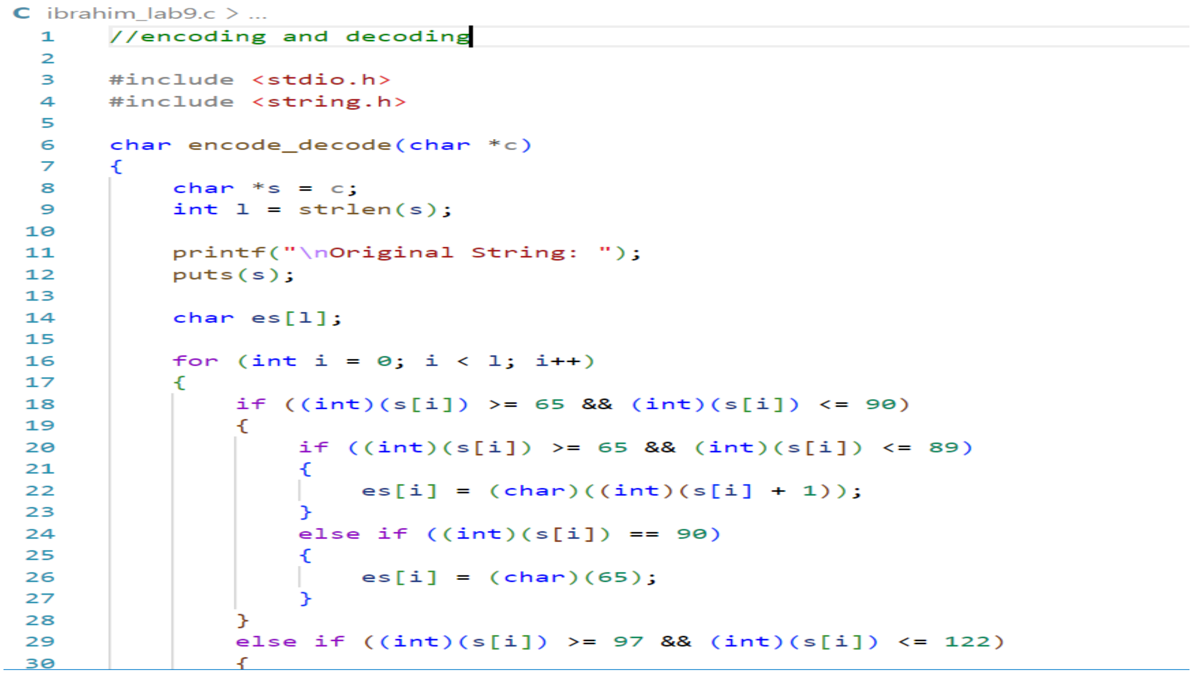
```c
        }
        i++;
    }
    if (substr[j] == '\0' && flag)
    {
        for (i = 0; i < start; i++)
            output[i] = str[i];

        for (j = 0; j < strlen(replace); j++) // replace substring with another string
        {
            output[i] = replace[j];
            i++;
        }

        for (j = start + strlen(substr); j < strlen(str); j++) // copying remaining portion of the input string
        {
            output[i] = str[j];
            i++;
        }

        output[i] = '\0';
        printf("Output: %s\n", output);
    }
    else
    {
        printf("%s is not a substring of %s\n", substr, str);
    }
    return 0;
}
```

Output: 8

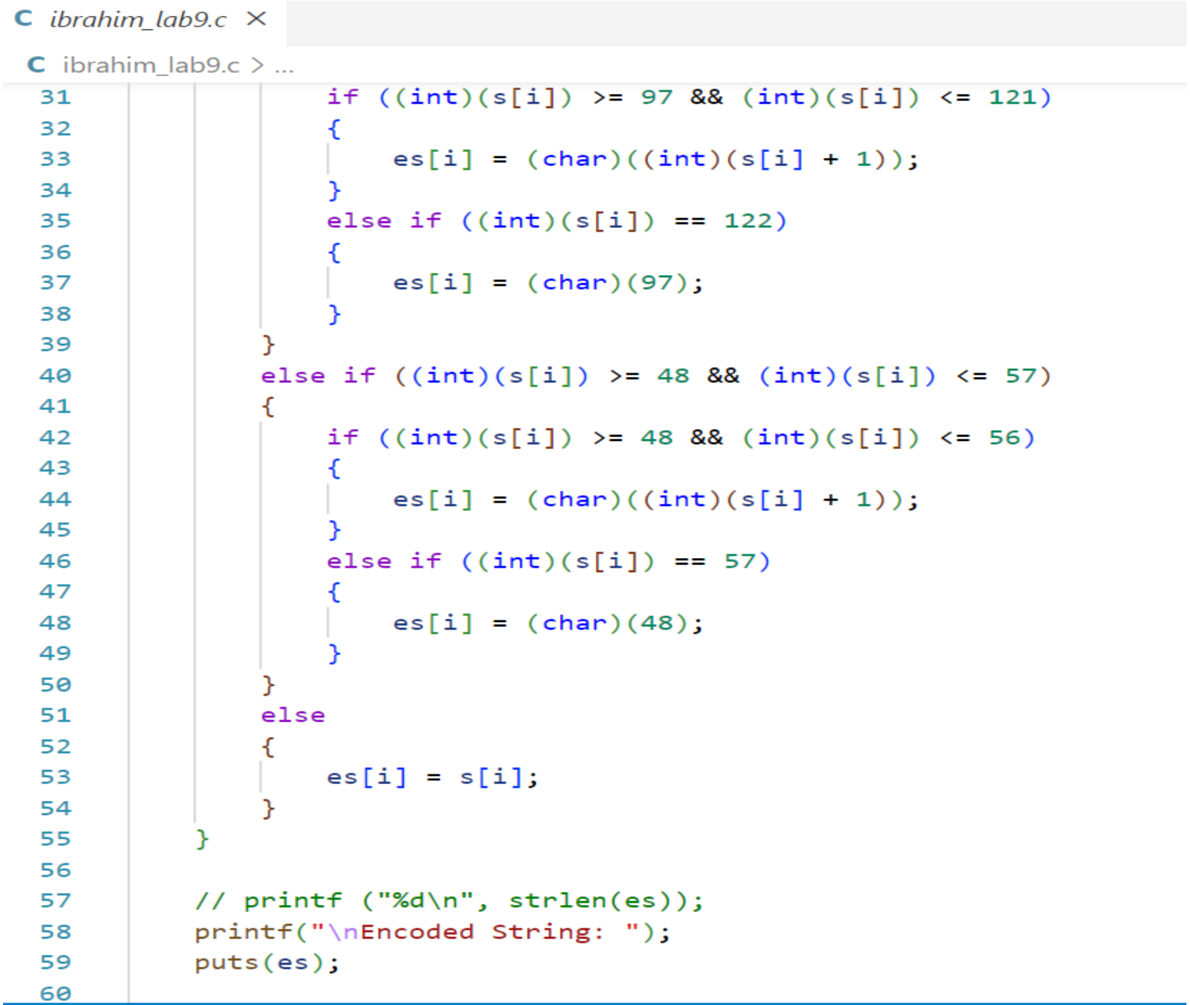


```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab8.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter the string: md-ibrahim
Enter the substring: ibrahim
Enter the string you want substring to get replaced with: akhtar
Output: md-akhtar
PS D:\Programming\C Lab Programs HW> 
```

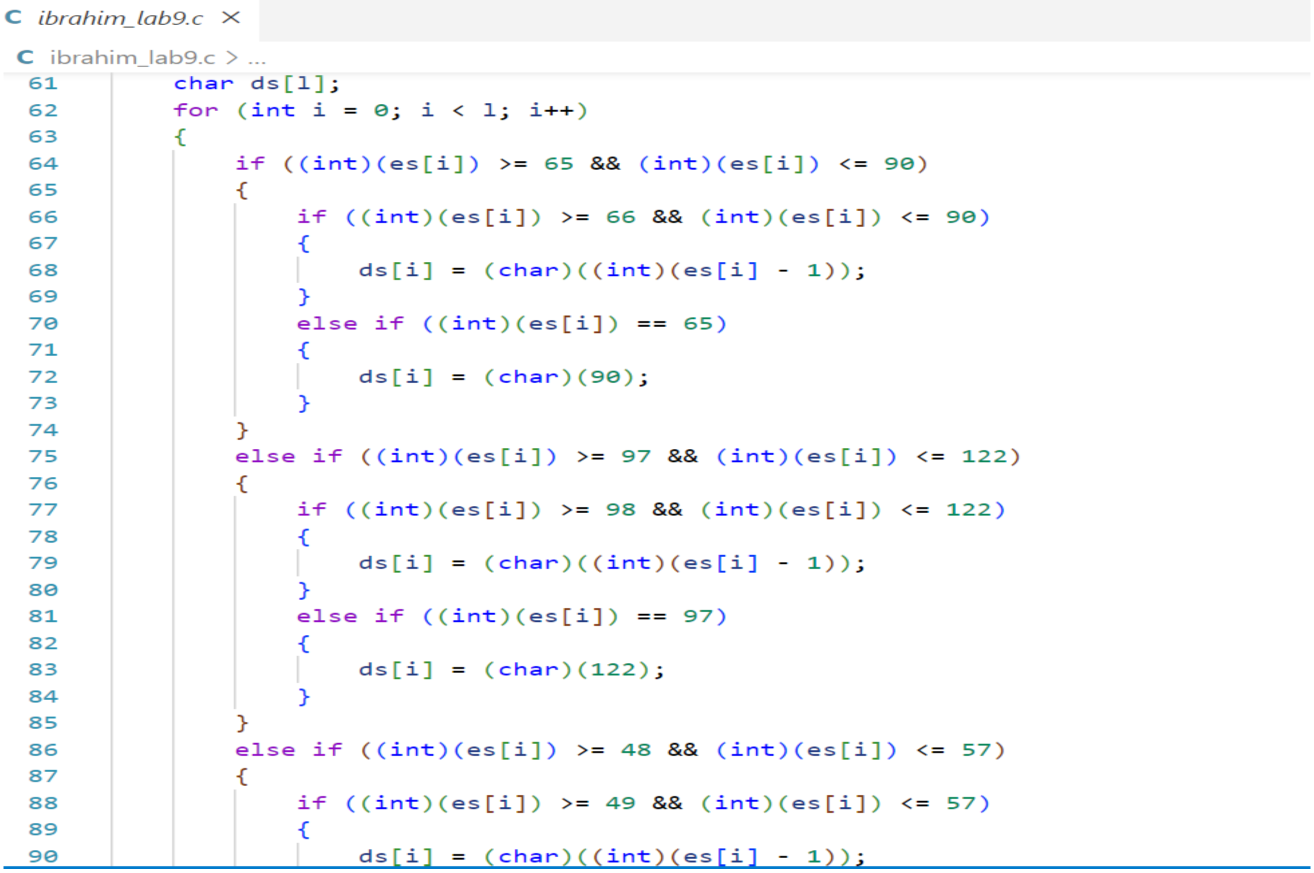
Program: 9



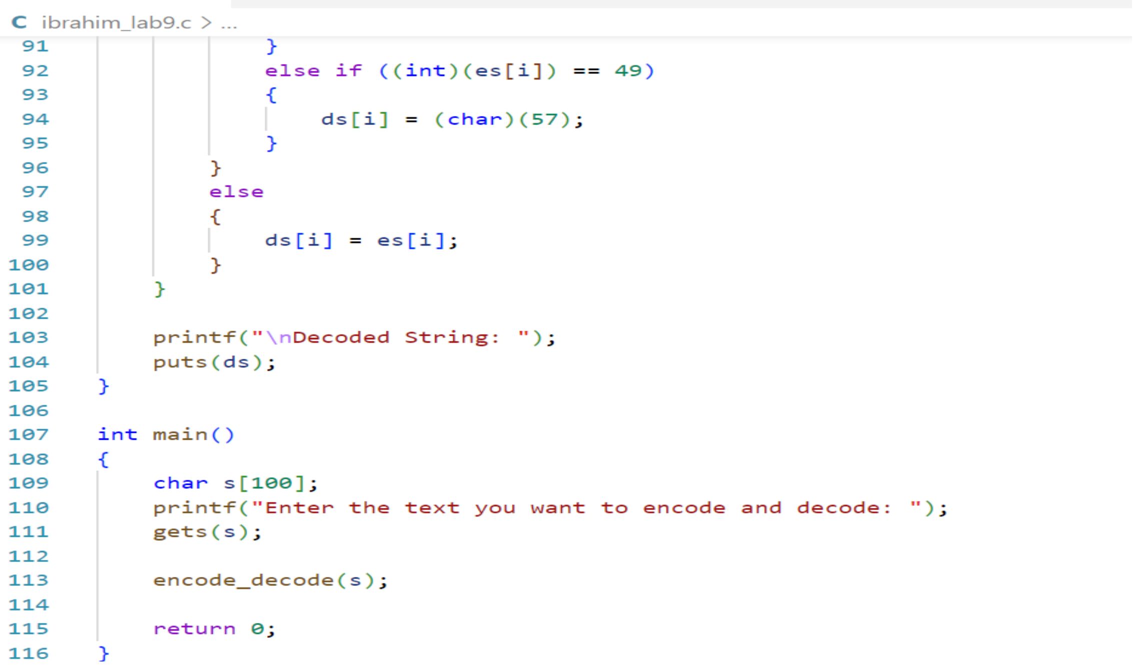
```c
//encoding and decoding

#include <stdio.h>
#include <string.h>

char encode_decode(char *c)
{
    char *s = c;
    int l = strlen(s);

    printf("\nOriginal String: ");
    puts(s);

    char es[l];

    for (int i = 0; i < l; i++)
    {
        if ((int)(s[i]) >= 65 && (int)(s[i]) <= 90)
        {
            if ((int)(s[i]) >= 65 && (int)(s[i]) <= 89)
            {
                es[i] = (char)((int)(s[i] + 1));
            }
            else if ((int)(s[i]) == 90)
            {
                es[i] = (char)(65);
            }
        }
        else if ((int)(s[i]) >= 97 && (int)(s[i]) <= 122)
        {
```



```
            if ((int)(s[i]) >= 97 && (int)(s[i]) <= 121)
            {
                es[i] = (char)((int)(s[i] + 1));
            }
            else if ((int)(s[i]) == 122)
            {
                es[i] = (char)(97);
            }
        }
        else if ((int)(s[i]) >= 48 && (int)(s[i]) <= 57)
        {
            if ((int)(s[i]) >= 48 && (int)(s[i]) <= 56)
            {
                es[i] = (char)((int)(s[i] + 1));
            }
            else if ((int)(s[i]) == 57)
            {
                es[i] = (char)(48);
            }
        }
        else
        {
            es[i] = s[i];
        }
    }

    // printf ("%d\n", strlen(es));
    printf("\nEncoded String: ");
    puts(es);
```

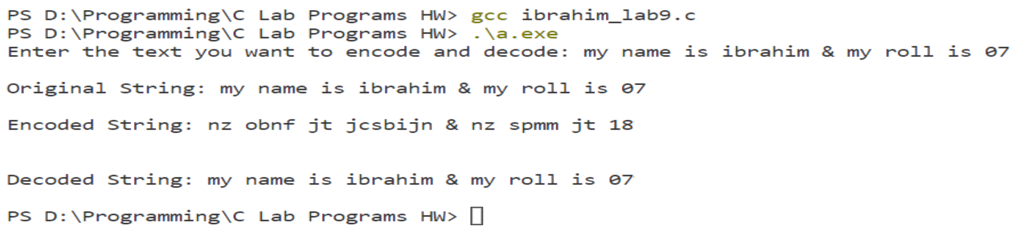


```
    char ds[l];
    for (int i = 0; i < l; i++)
    {
        if ((int)(es[i]) >= 65 && (int)(es[i]) <= 90)
        {
            if ((int)(es[i]) >= 66 && (int)(es[i]) <= 90)
            {
                ds[i] = (char)((int)(es[i] - 1));
            }
            else if ((int)(es[i]) == 65)
            {
                ds[i] = (char)(90);
            }
        }
        else if ((int)(es[i]) >= 97 && (int)(es[i]) <= 122)
        {
            if ((int)(es[i]) >= 98 && (int)(es[i]) <= 122)
            {
                ds[i] = (char)((int)(es[i] - 1));
            }
            else if ((int)(es[i]) == 97)
            {
                ds[i] = (char)(122);
            }
        }
        else if ((int)(es[i]) >= 48 && (int)(es[i]) <= 57)
        {
            if ((int)(es[i]) >= 49 && (int)(es[i]) <= 57)
            {
                ds[i] = (char)((int)(es[i] - 1));
```



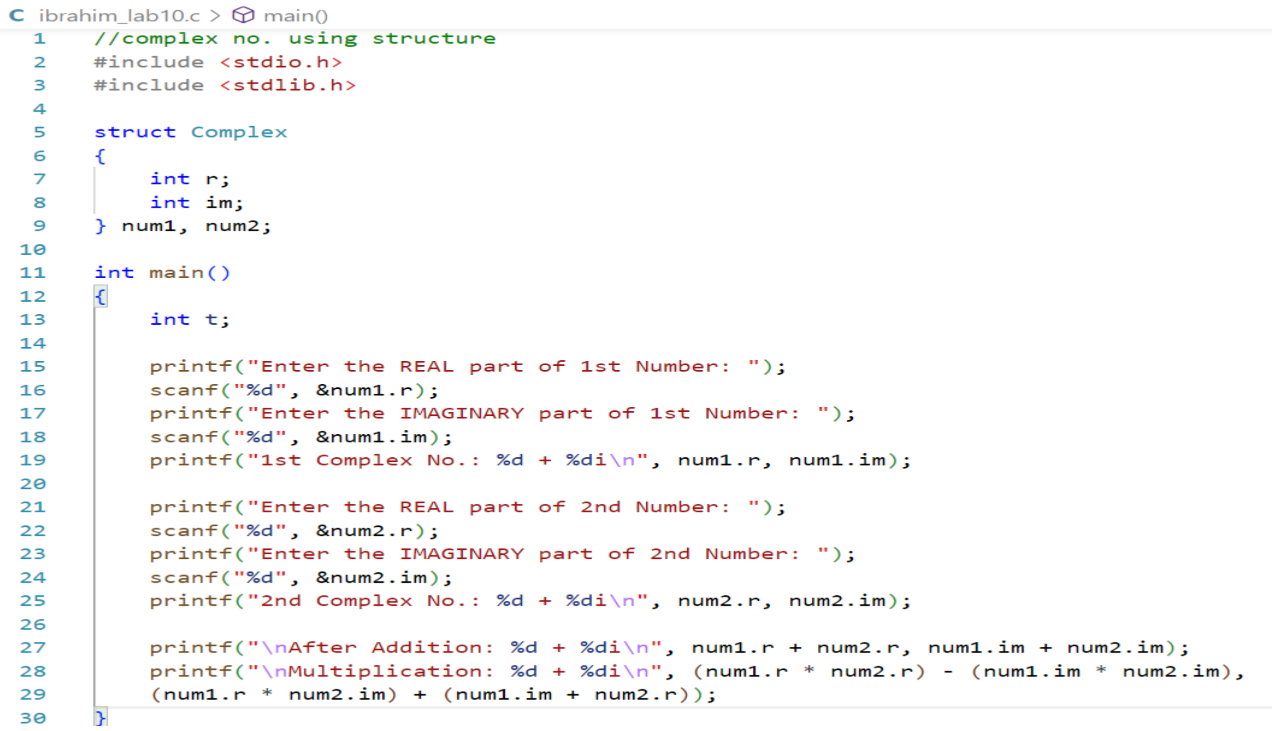
```c
            }
            else if ((int)(es[i]) == 49)
            {
                ds[i] = (char)(57);
            }
        }
        else
        {
            ds[i] = es[i];
        }
    }

    printf("\nDecoded String: ");
    puts(ds);
}

int main()
{
    char s[100];
    printf("Enter the text you want to encode and decode: ");
    gets(s);

    encode_decode(s);

    return 0;
}
```

Output: 9



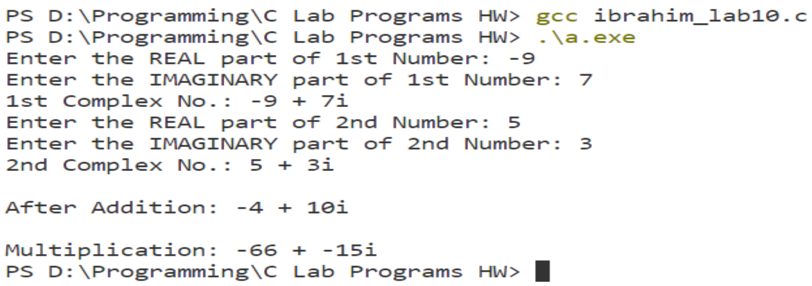
```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab9.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter the text you want to encode and decode: my name is ibrahim & my roll is 07

Original String: my name is ibrahim & my roll is 07

Encoded String: nz obnf jt jcsbijn & nz spmm jt 18


Decoded String: my name is ibrahim & my roll is 07

PS D:\Programming\C Lab Programs HW> 
```

Program: 10



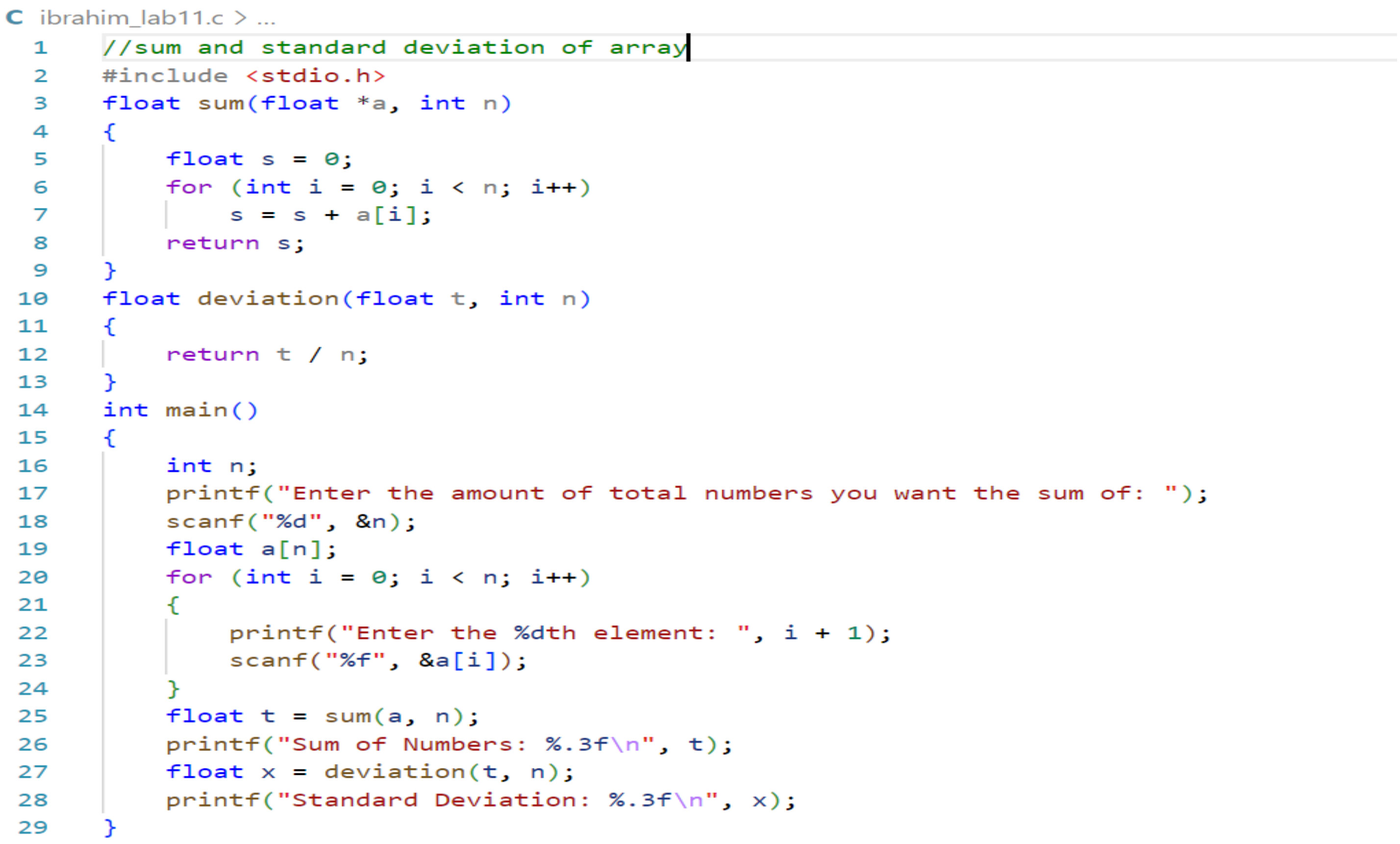
```c
//complex no. using structure
#include <stdio.h>
#include <stdlib.h>

struct Complex
{
    int r;
    int im;
} num1, num2;

int main()
{
    int t;

    printf("Enter the REAL part of 1st Number: ");
    scanf("%d", &num1.r);
    printf("Enter the IMAGINARY part of 1st Number: ");
    scanf("%d", &num1.im);
    printf("1st Complex No.: %d + %di\n", num1.r, num1.im);

    printf("Enter the REAL part of 2nd Number: ");
    scanf("%d", &num2.r);
    printf("Enter the IMAGINARY part of 2nd Number: ");
    scanf("%d", &num2.im);
    printf("2nd Complex No.: %d + %di\n", num2.r, num2.im);

    printf("\nAfter Addition: %d + %di\n", num1.r + num2.r, num1.im + num2.im);
    printf("\nMultiplication: %d + %di\n", (num1.r * num2.r) - (num1.im * num2.im),
    (num1.r * num2.im) + (num1.im + num2.r));
}
```

Output: 10



```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab10.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter the REAL part of 1st Number: -9
Enter the IMAGINARY part of 1st Number: 7
1st Complex No.: -9 + 7i
Enter the REAL part of 2nd Number: 5
Enter the IMAGINARY part of 2nd Number: 3
2nd Complex No.: 5 + 3i

After Addition: -4 + 10i

Multiplication: -66 + -15i
PS D:\Programming\C Lab Programs HW> 
```

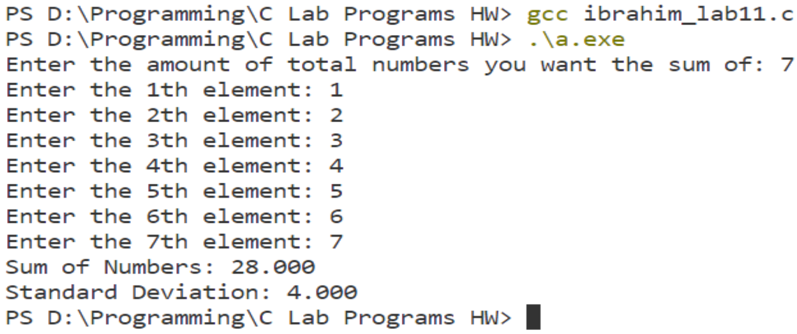
Program: 11



C ibrahim_lab11.c > ...

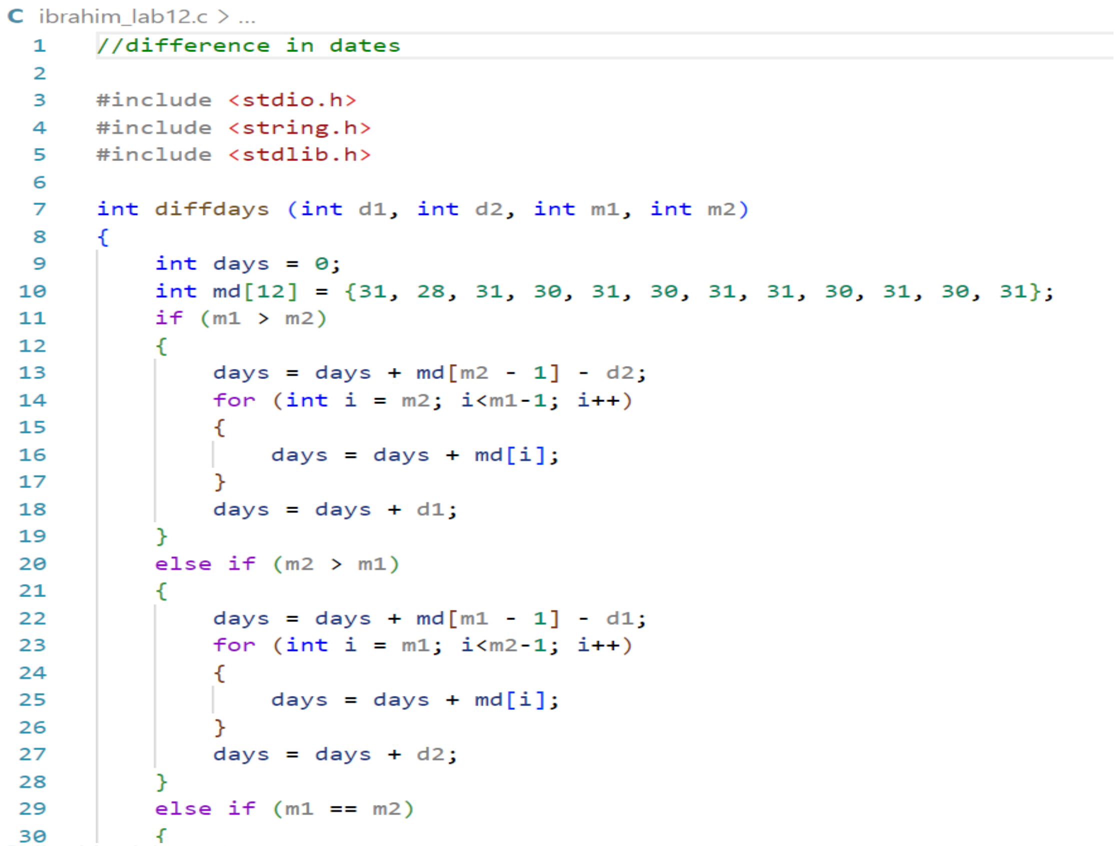
```c
//sum and standard deviation of array
#include <stdio.h>
float sum(float *a, int n)
{
    float s = 0;
    for (int i = 0; i < n; i++)
        s = s + a[i];
    return s;
}
float deviation(float t, int n)
{
    return t / n;
}
int main()
{
    int n;
    printf("Enter the amount of total numbers you want the sum of: ");
    scanf("%d", &n);
    float a[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter the %dth element: ", i + 1);
        scanf("%f", &a[i]);
    }
    float t = sum(a, n);
    printf("Sum of Numbers: %.3f\n", t);
    float x = deviation(t, n);
    printf("Standard Deviation: %.3f\n", x);
}
```

Output: 11

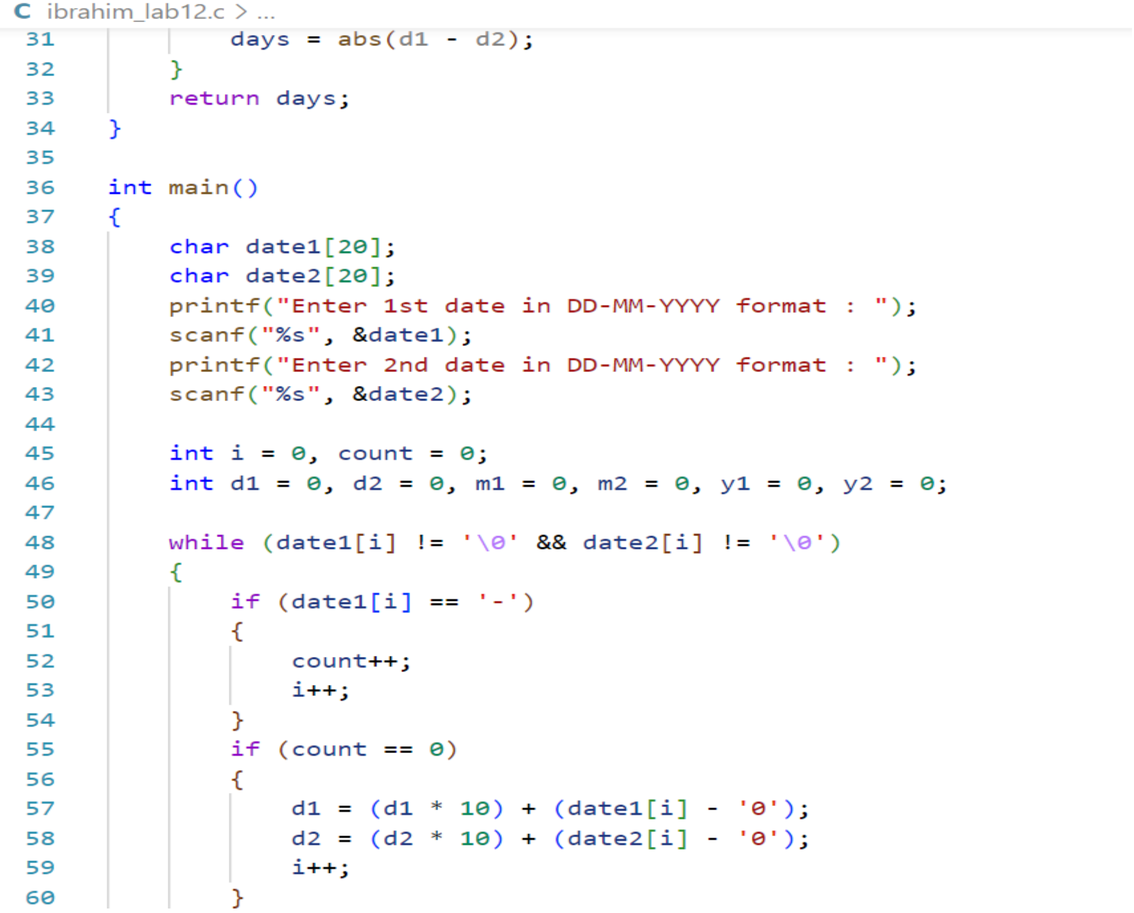


```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab11.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter the amount of total numbers you want the sum of: 7
Enter the 1th element: 1
Enter the 2th element: 2
Enter the 3th element: 3
Enter the 4th element: 4
Enter the 5th element: 5
Enter the 6th element: 6
Enter the 7th element: 7
Sum of Numbers: 28.000
Standard Deviation: 4.000
PS D:\Programming\C Lab Programs HW> 
```

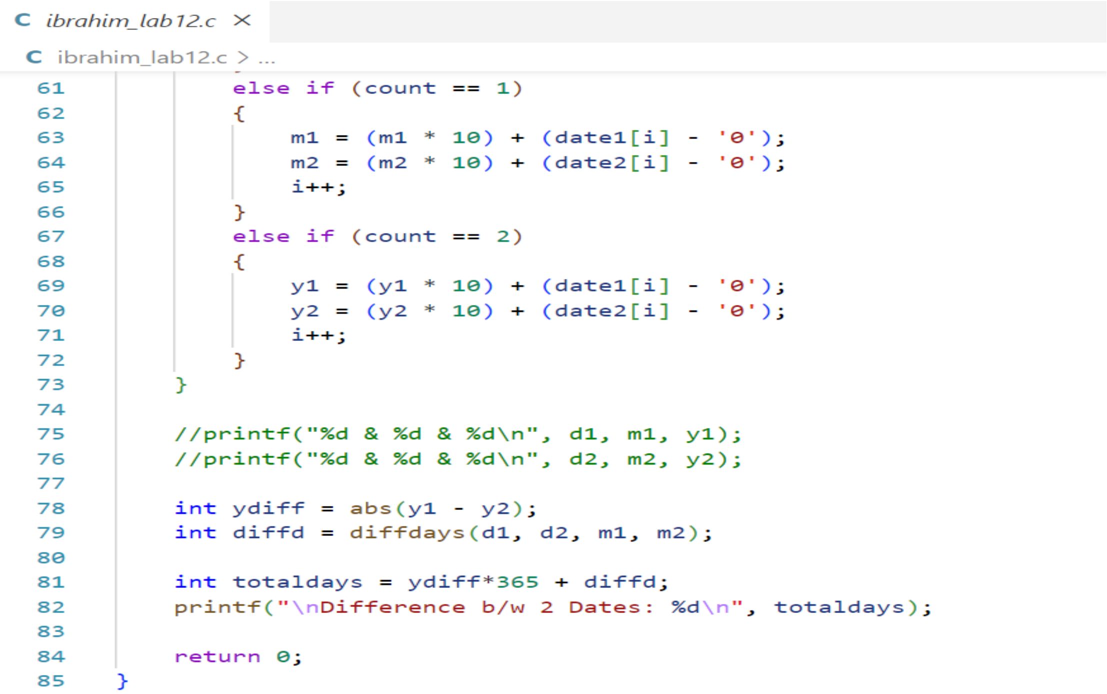
Program: 12



```c
//difference in dates

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int diffdays (int d1, int d2, int m1, int m2)
{
    int days = 0;
    int md[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if (m1 > m2)
    {
        days = days + md[m2 - 1] - d2;
        for (int i = m2; i<m1-1; i++)
        {
            days = days + md[i];
        }
        days = days + d1;
    }
    else if (m2 > m1)
    {
        days = days + md[m1 - 1] - d1;
        for (int i = m1; i<m2-1; i++)
        {
            days = days + md[i];
        }
        days = days + d2;
    }
    else if (m1 == m2)
    {
```

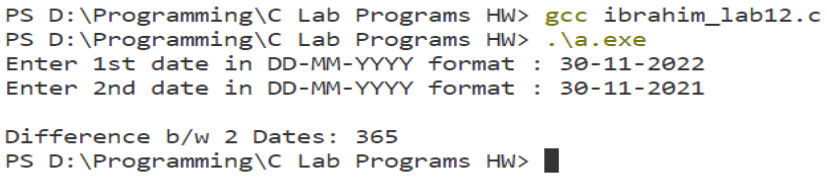


```c
        days = abs(d1 - d2);
    }
    return days;
}

int main()
{
    char date1[20];
    char date2[20];
    printf("Enter 1st date in DD-MM-YYYY format : ");
    scanf("%s", &date1);
    printf("Enter 2nd date in DD-MM-YYYY format : ");
    scanf("%s", &date2);

    int i = 0, count = 0;
    int d1 = 0, d2 = 0, m1 = 0, m2 = 0, y1 = 0, y2 = 0;

    while (date1[i] != '\0' && date2[i] != '\0')
    {
        if (date1[i] == '-')
        {
            count++;
            i++;
        }
        if (count == 0)
        {
            d1 = (d1 * 10) + (date1[i] - '0');
            d2 = (d2 * 10) + (date2[i] - '0');
            i++;
        }
```



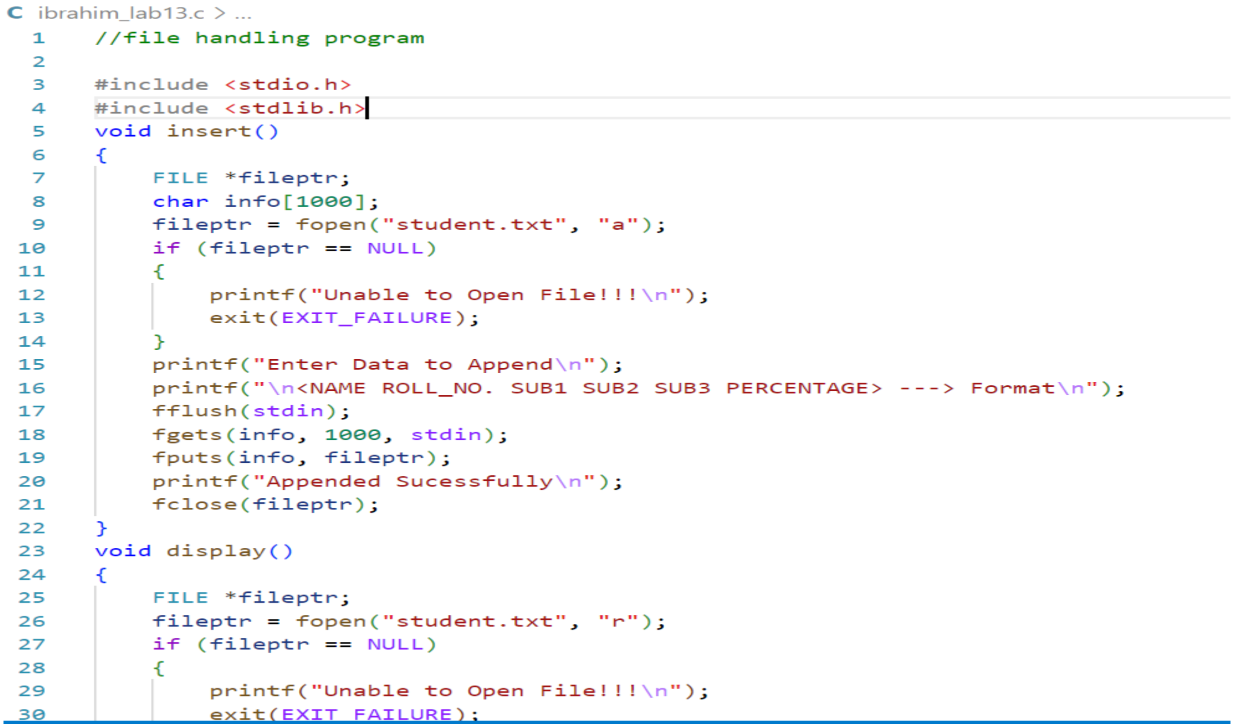
```c
        else if (count == 1)
        {
            m1 = (m1 * 10) + (date1[i] - '0');
            m2 = (m2 * 10) + (date2[i] - '0');
            i++;
        }
        else if (count == 2)
        {
            y1 = (y1 * 10) + (date1[i] - '0');
            y2 = (y2 * 10) + (date2[i] - '0');
            i++;
        }
    }

    //printf("%d & %d & %d\n", d1, m1, y1);
    //printf("%d & %d & %d\n", d2, m2, y2);

    int ydiff = abs(y1 - y2);
    int diffd = diffdays(d1, d2, m1, m2);

    int totaldays = ydiff*365 + diffd;
    printf("\nDifference b/w 2 Dates: %d\n", totaldays);

    return 0;
}
```

Output: 12

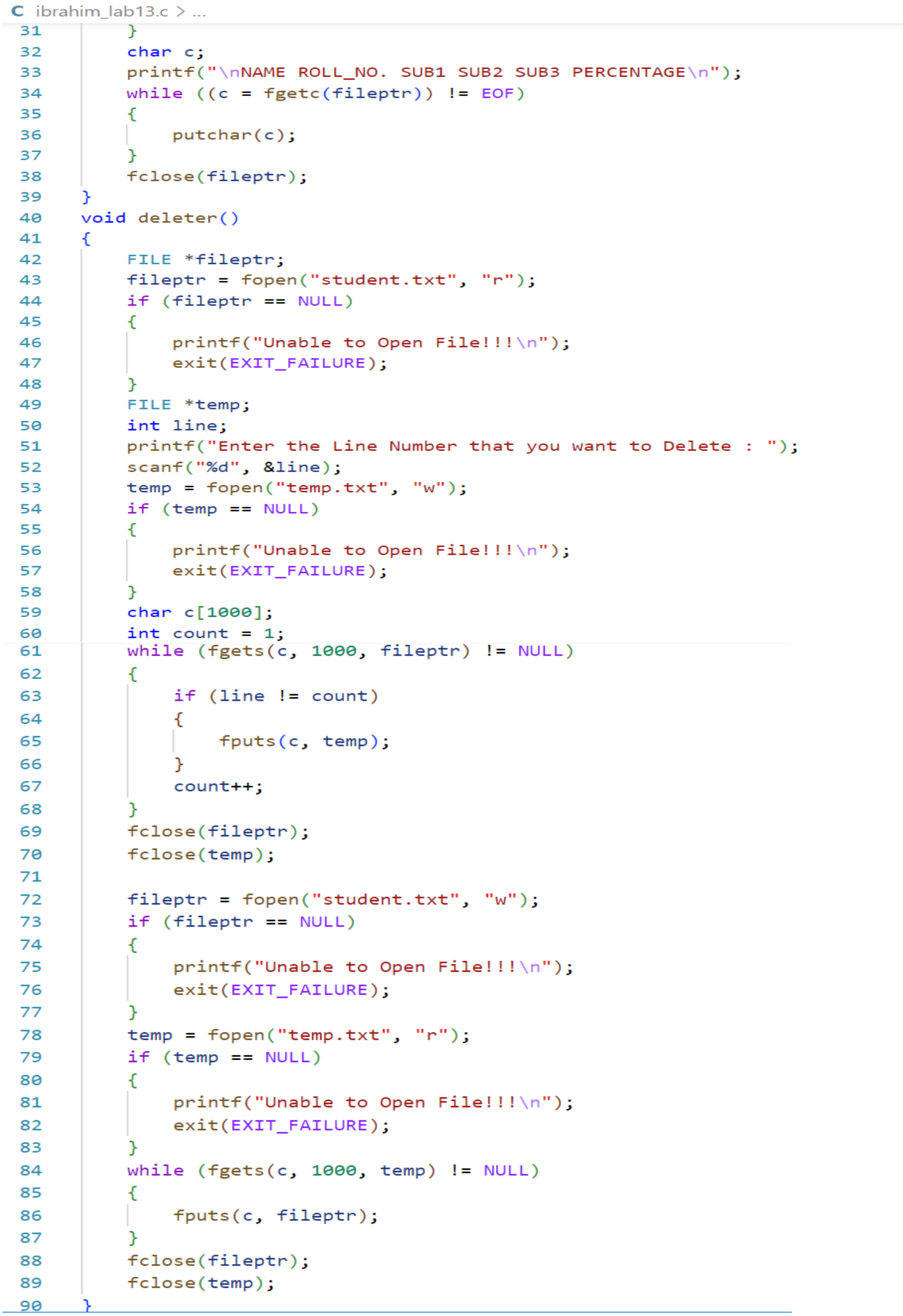


```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab12.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter 1st date in DD-MM-YYYY format : 30-11-2022
Enter 2nd date in DD-MM-YYYY format : 30-11-2021

Difference b/w 2 Dates: 365
PS D:\Programming\C Lab Programs HW> █
```

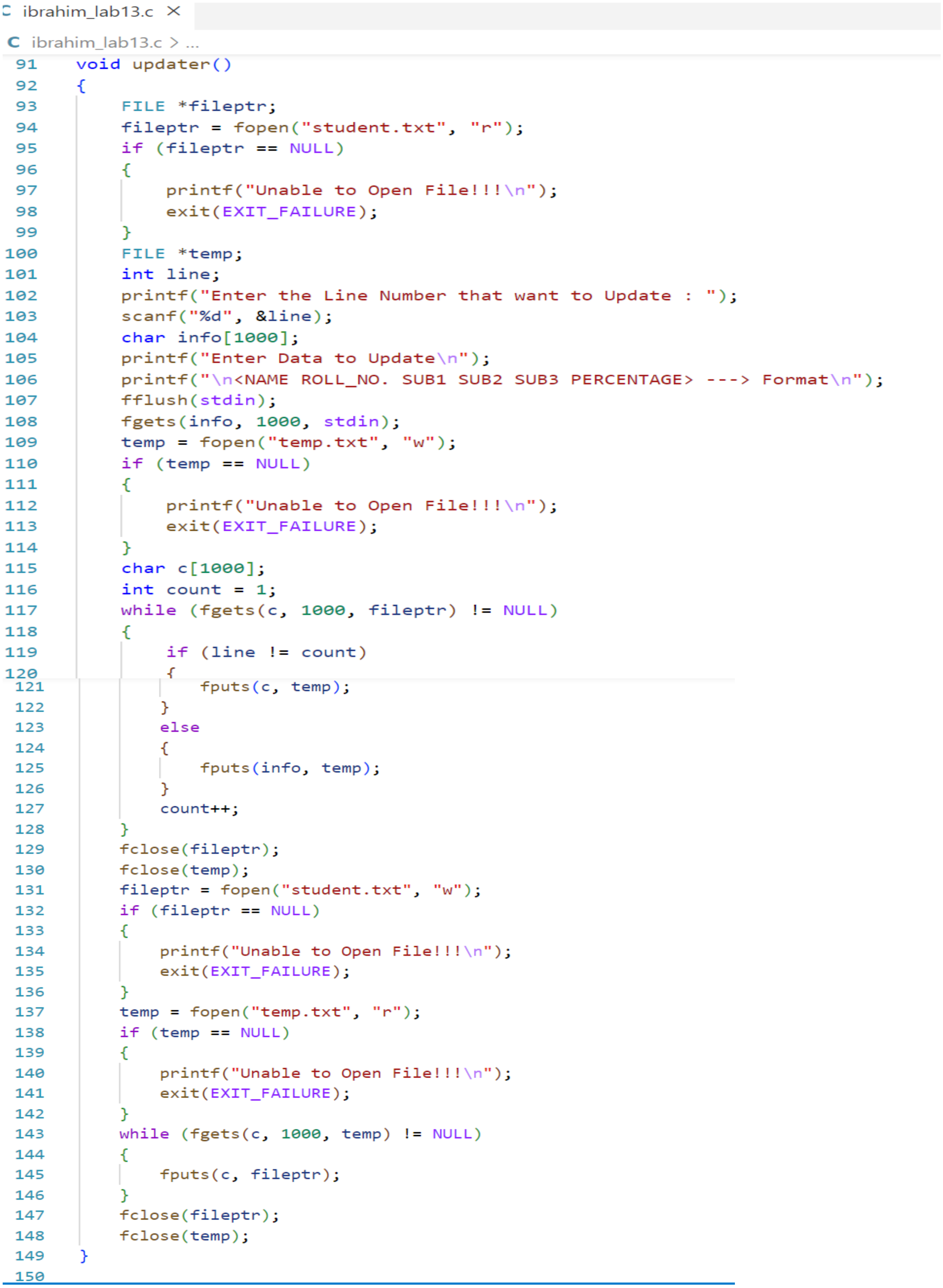
Program: 13



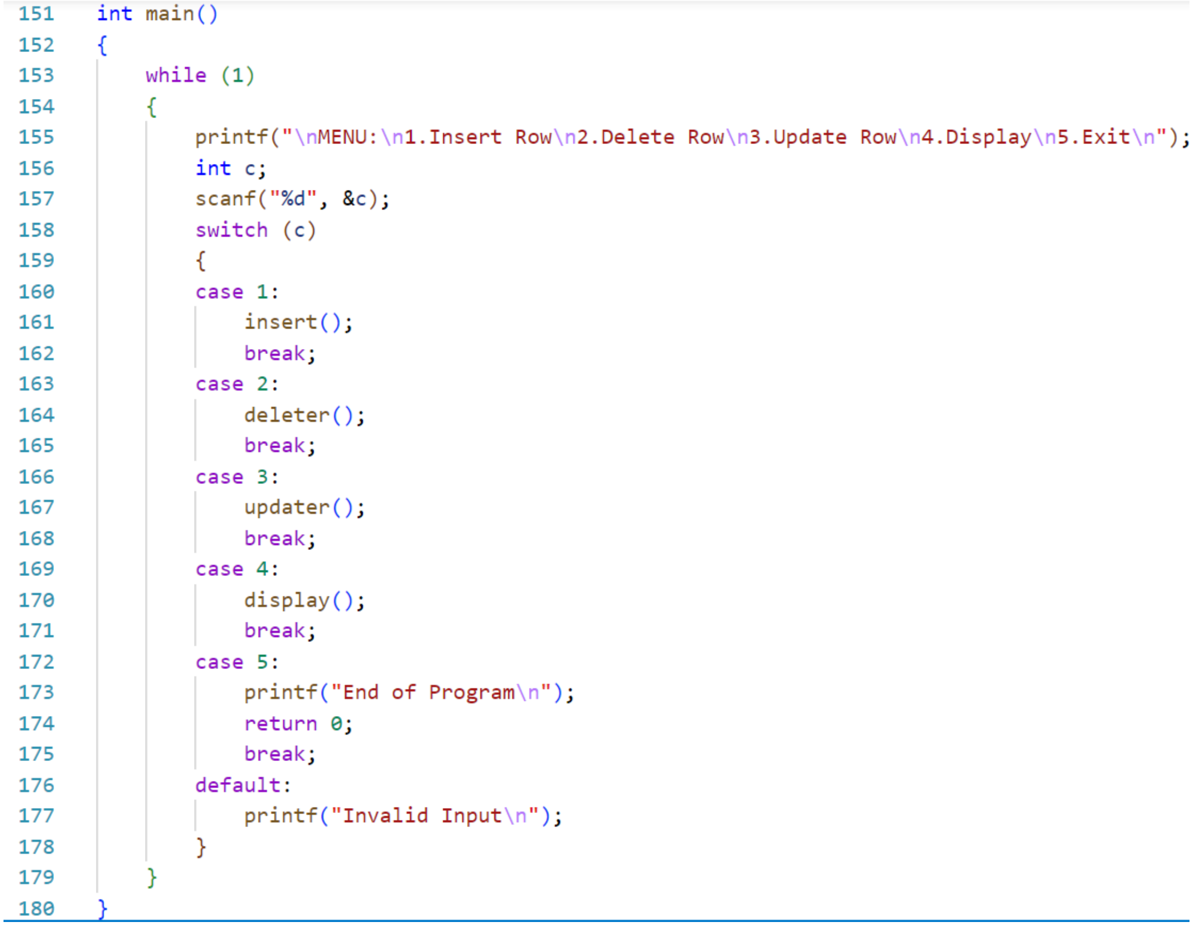
```c
//file handling program

#include <stdio.h>
#include <stdlib.h>
void insert()
{
    FILE *fileptr;
    char info[1000];
    fileptr = fopen("student.txt", "a");
    if (fileptr == NULL)
    {
        printf("Unable to Open File!!!\n");
        exit(EXIT_FAILURE);
    }
    printf("Enter Data to Append\n");
    printf("\n<NAME ROLL_NO. SUB1 SUB2 SUB3 PERCENTAGE> ---> Format\n");
    fflush(stdin);
    fgets(info, 1000, stdin);
    fputs(info, fileptr);
    printf("Appended Sucessfully\n");
    fclose(fileptr);
}
void display()
{
    FILE *fileptr;
    fileptr = fopen("student.txt", "r");
    if (fileptr == NULL)
    {
        printf("Unable to Open File!!!\n");
        exit(EXIT_FAILURE);
```



```
    }
    char c;
    printf("\nNAME ROLL_NO. SUB1 SUB2 SUB3 PERCENTAGE\n");
    while ((c = fgetc(fileptr)) != EOF)
    {
        putchar(c);
    }
    fclose(fileptr);
}
void deleter()
{
    FILE *fileptr;
    fileptr = fopen("student.txt", "r");
    if (fileptr == NULL)
    {
        printf("Unable to Open File!!!\n");
        exit(EXIT_FAILURE);
    }
    FILE *temp;
    int line;
    printf("Enter the Line Number that you want to Delete : ");
    scanf("%d", &line);
    temp = fopen("temp.txt", "w");
    if (temp == NULL)
    {
        printf("Unable to Open File!!!\n");
        exit(EXIT_FAILURE);
    }
    char c[1000];
    int count = 1;
    while (fgets(c, 1000, fileptr) != NULL)
    {
        if (line != count)
        {
            fputs(c, temp);
        }
        count++;
    }
    fclose(fileptr);
    fclose(temp);

    fileptr = fopen("student.txt", "w");
    if (fileptr == NULL)
    {
        printf("Unable to Open File!!!\n");
        exit(EXIT_FAILURE);
    }
    temp = fopen("temp.txt", "r");
    if (temp == NULL)
    {
        printf("Unable to Open File!!!\n");
        exit(EXIT_FAILURE);
    }
    while (fgets(c, 1000, temp) != NULL)
    {
        fputs(c, fileptr);
    }
    fclose(fileptr);
    fclose(temp);
}
```

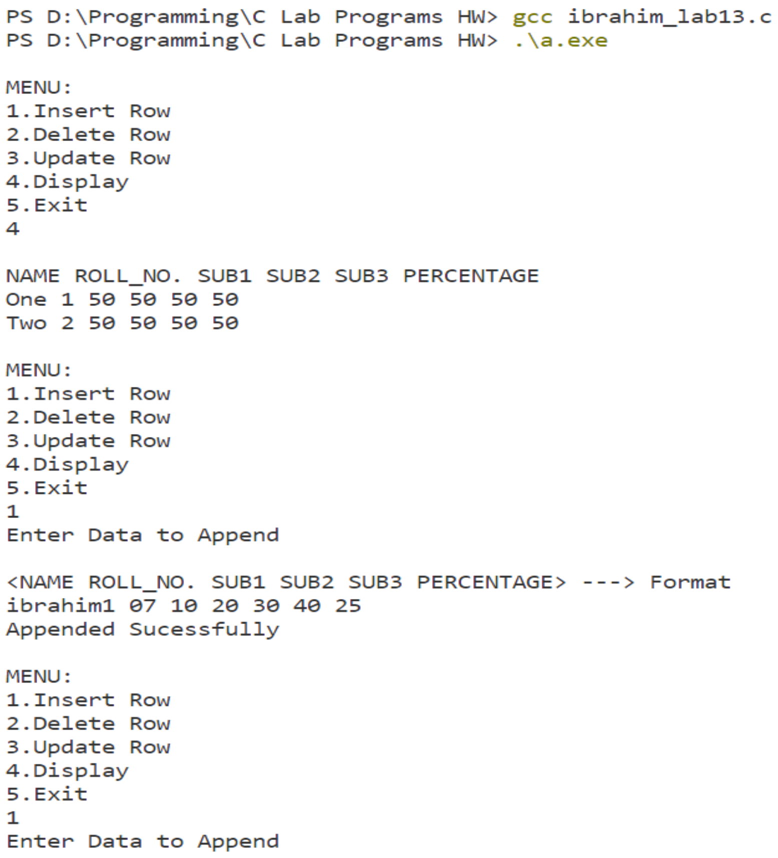


```c
void updater()
{
    FILE *fileptr;
    fileptr = fopen("student.txt", "r");
    if (fileptr == NULL)
    {
        printf("Unable to Open File!!!\n");
        exit(EXIT_FAILURE);
    }
    FILE *temp;
    int line;
    printf("Enter the Line Number that want to Update : ");
    scanf("%d", &line);
    char info[1000];
    printf("Enter Data to Update\n");
    printf("\n<NAME ROLL_NO. SUB1 SUB2 SUB3 PERCENTAGE> ---> Format\n");
    fflush(stdin);
    fgets(info, 1000, stdin);
    temp = fopen("temp.txt", "w");
    if (temp == NULL)
    {
        printf("Unable to Open File!!!\n");
        exit(EXIT_FAILURE);
    }
    char c[1000];
    int count = 1;
    while (fgets(c, 1000, fileptr) != NULL)
    {
        if (line != count)
        {
            fputs(c, temp);
        }
        else
        {
            fputs(info, temp);
        }
        count++;
    }
    fclose(fileptr);
    fclose(temp);
    fileptr = fopen("student.txt", "w");
    if (fileptr == NULL)
    {
        printf("Unable to Open File!!!\n");
        exit(EXIT_FAILURE);
    }
    temp = fopen("temp.txt", "r");
    if (temp == NULL)
    {
        printf("Unable to Open File!!!\n");
        exit(EXIT_FAILURE);
    }
    while (fgets(c, 1000, temp) != NULL)
    {
        fputs(c, fileptr);
    }
    fclose(fileptr);
    fclose(temp);
}
```

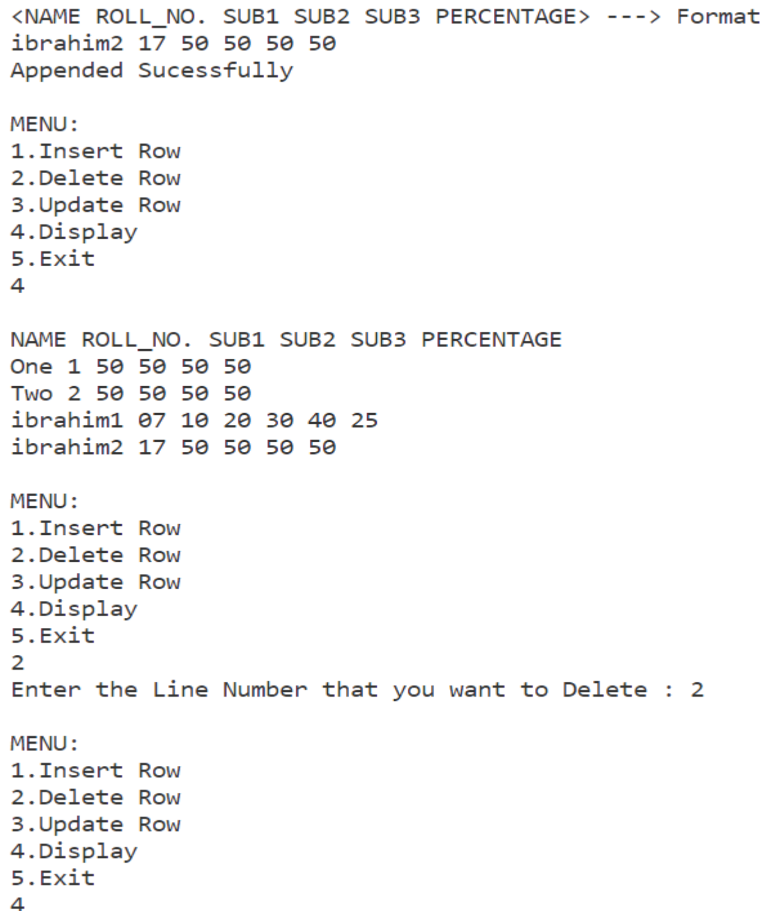


```c
int main()
{
    while (1)
    {
        printf("\nMENU:\n1.Insert Row\n2.Delete Row\n3.Update Row\n4.Display\n5.Exit\n");
        int c;
        scanf("%d", &c);
        switch (c)
        {
        case 1:
            insert();
            break;
        case 2:
            deleter();
            break;
        case 3:
            updater();
            break;
        case 4:
            display();
            break;
        case 5:
            printf("End of Program\n");
            return 0;
            break;
        default:
            printf("Invalid Input\n");
        }
    }
}
```

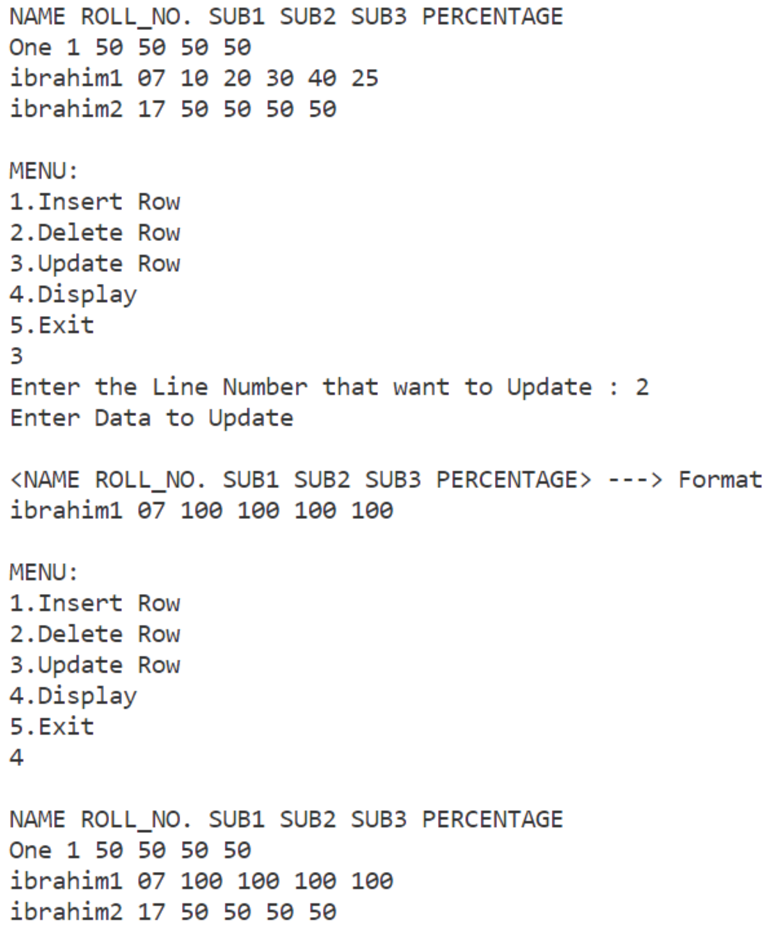
Output: 13



```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab13.c
PS D:\Programming\C Lab Programs HW> .\a.exe

MENU:
1.Insert Row
2.Delete Row
3.Update Row
4.Display
5.Exit
4

NAME ROLL_NO. SUB1 SUB2 SUB3 PERCENTAGE
One 1 50 50 50 50
Two 2 50 50 50 50

MENU:
1.Insert Row
2.Delete Row
3.Update Row
4.Display
5.Exit
1
Enter Data to Append

<NAME ROLL_NO. SUB1 SUB2 SUB3 PERCENTAGE> ---> Format
ibrahim1 07 10 20 30 40 25
Appended Sucessfully

MENU:
1.Insert Row
2.Delete Row
3.Update Row
4.Display
5.Exit
1
Enter Data to Append
```

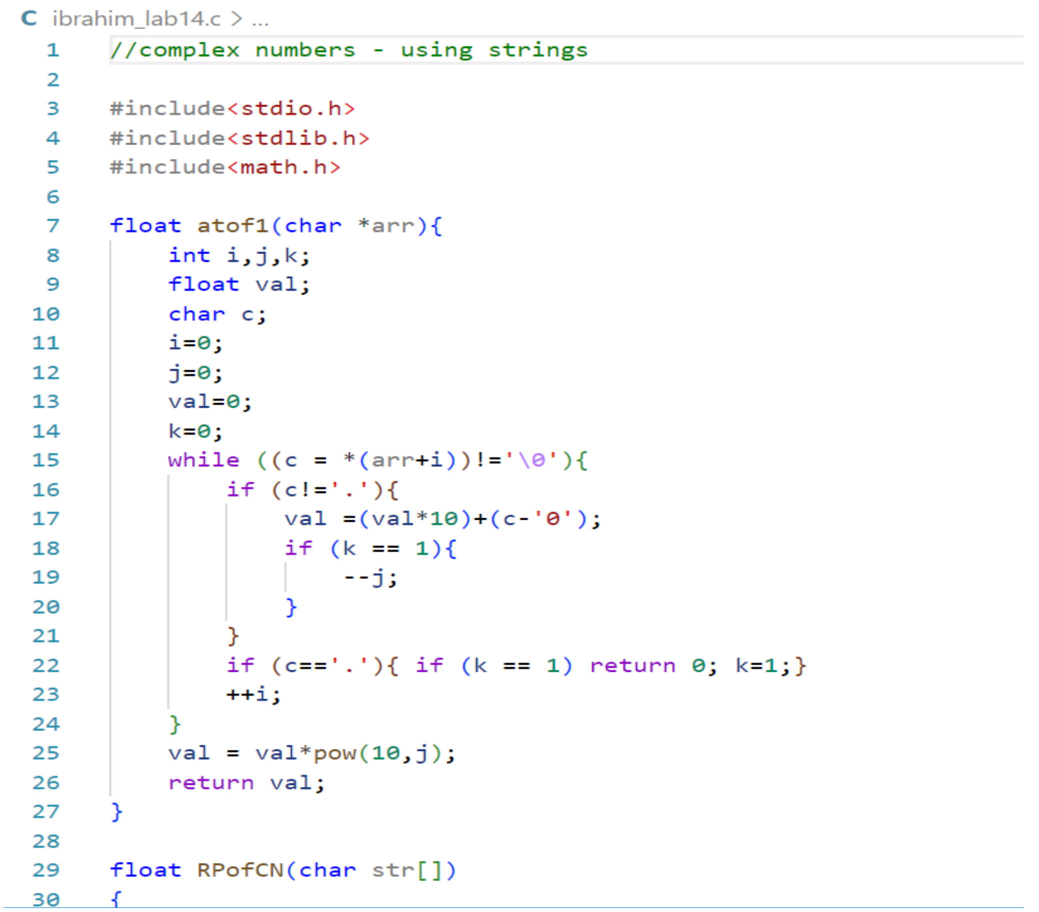


```
<NAME ROLL_NO. SUB1 SUB2 SUB3 PERCENTAGE> ---> Format
ibrahim2 17 50 50 50 50
Appended Sucessfully

MENU:
1.Insert Row
2.Delete Row
3.Update Row
4.Display
5.Exit
4

NAME ROLL_NO. SUB1 SUB2 SUB3 PERCENTAGE
One 1 50 50 50 50
Two 2 50 50 50 50
ibrahim1 07 10 20 30 40 25
ibrahim2 17 50 50 50 50

MENU:
1.Insert Row
2.Delete Row
3.Update Row
4.Display
5.Exit
2
Enter the Line Number that you want to Delete : 2

MENU:
1.Insert Row
2.Delete Row
3.Update Row
4.Display
5.Exit
4
```

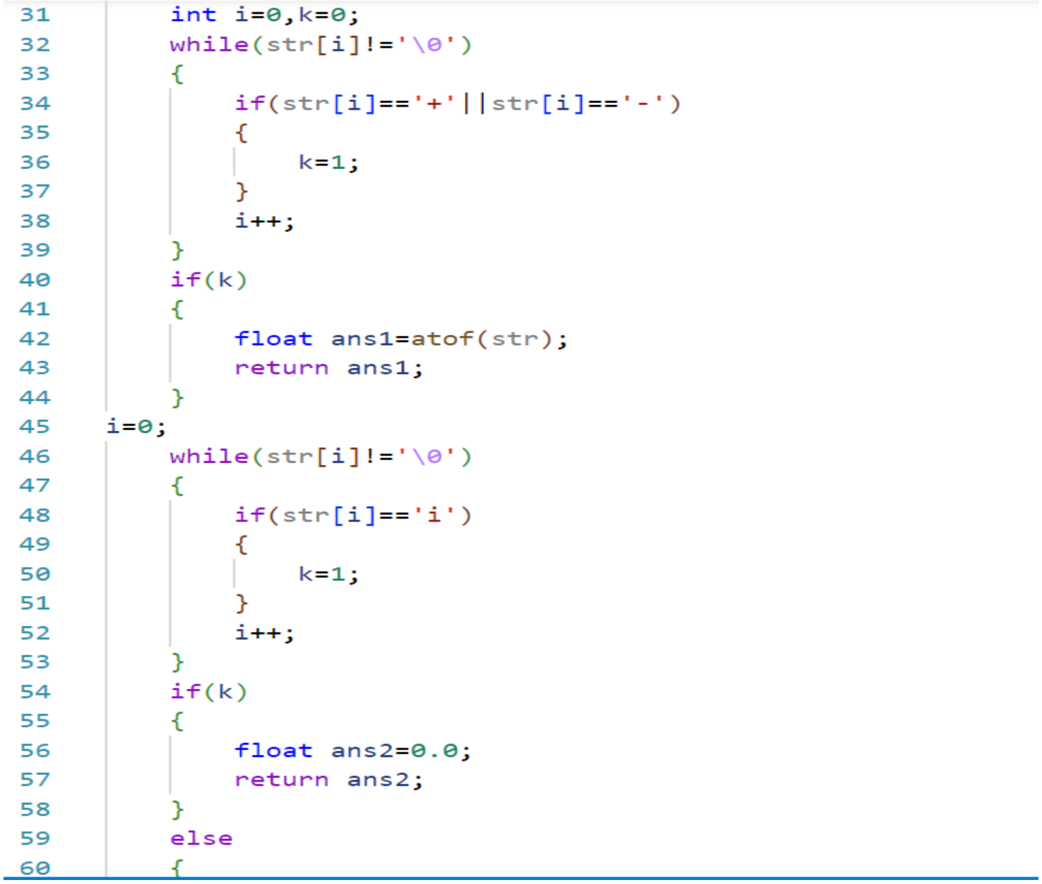


```
NAME ROLL_NO. SUB1 SUB2 SUB3 PERCENTAGE
One 1 50 50 50 50
ibrahim1 07 10 20 30 40 25
ibrahim2 17 50 50 50 50

MENU:
1.Insert Row
2.Delete Row
3.Update Row
4.Display
5.Exit
3
Enter the Line Number that want to Update : 2
Enter Data to Update

<NAME ROLL_NO. SUB1 SUB2 SUB3 PERCENTAGE> ---> Format
ibrahim1 07 100 100 100 100

MENU:
1.Insert Row
2.Delete Row
3.Update Row
4.Display
5.Exit
4

NAME ROLL_NO. SUB1 SUB2 SUB3 PERCENTAGE
One 1 50 50 50 50
ibrahim1 07 100 100 100 100
ibrahim2 17 50 50 50 50
```

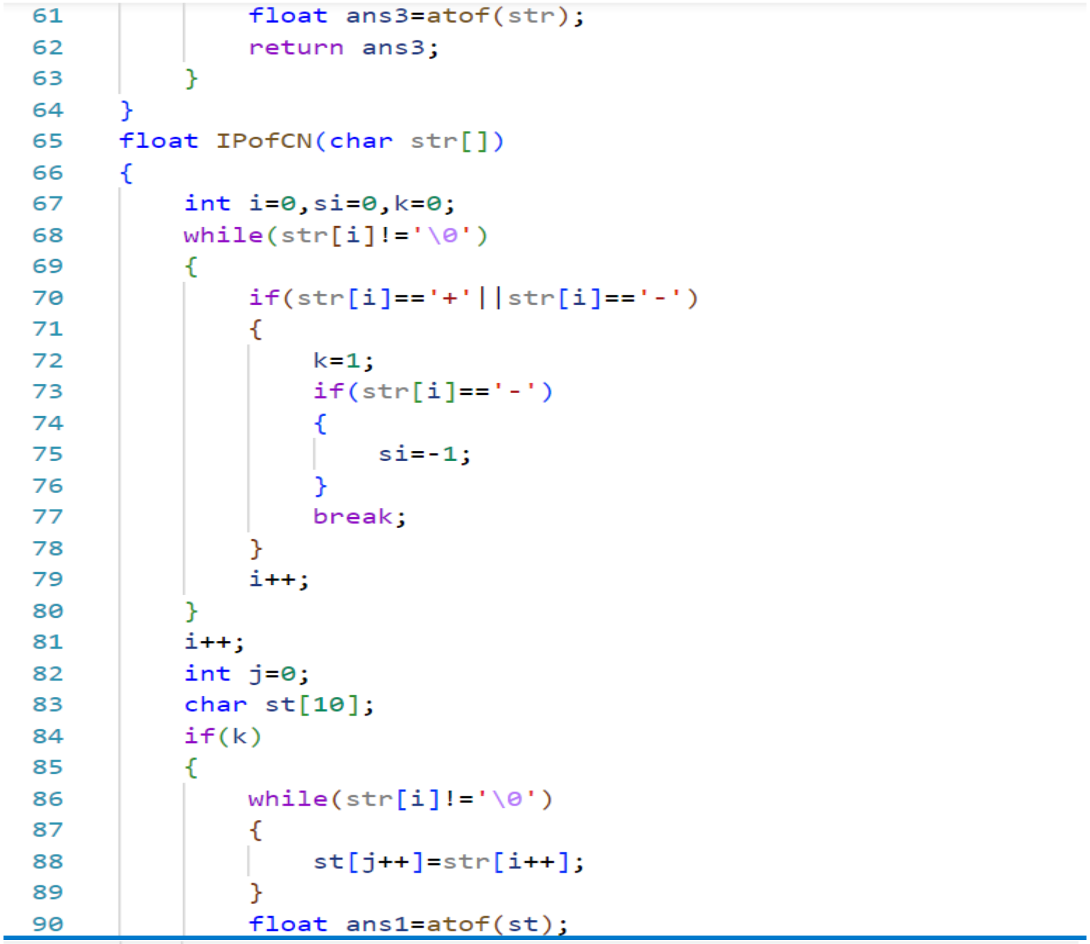
Program: 14



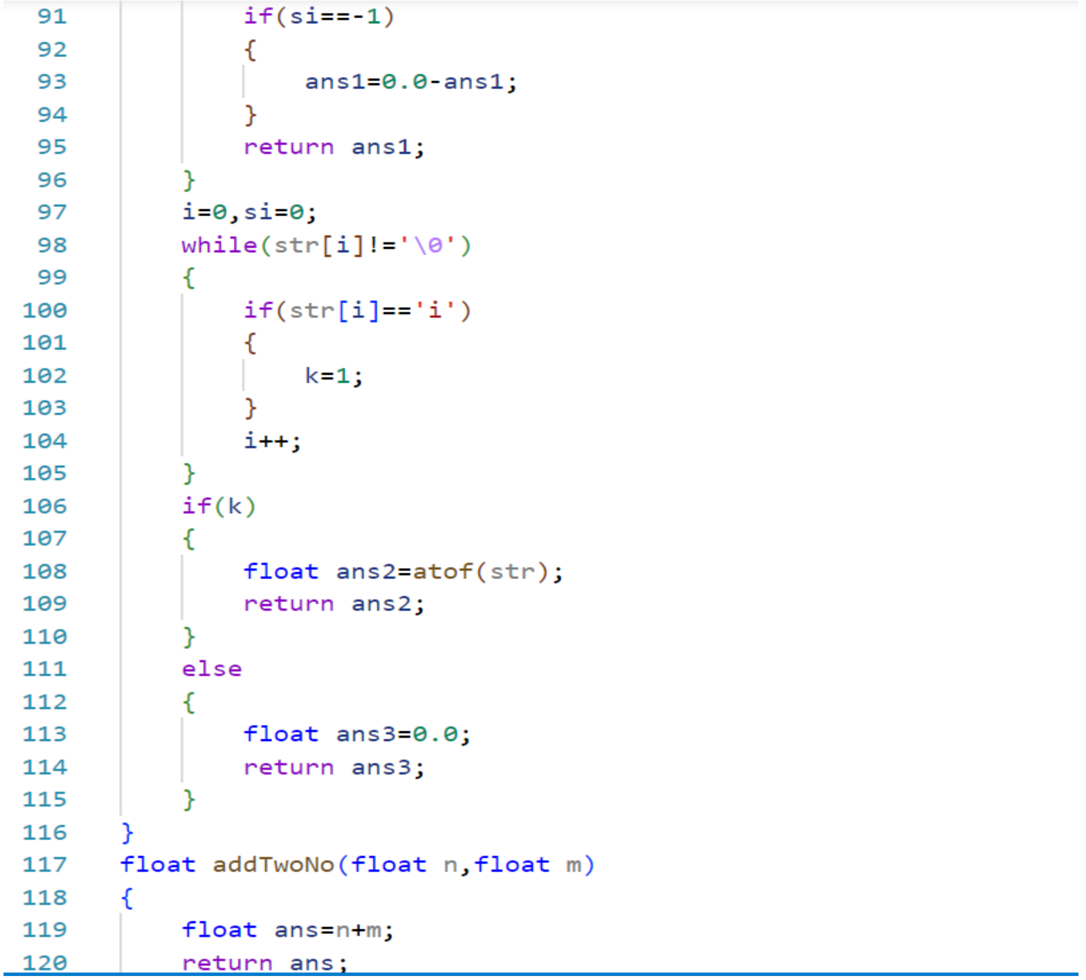
```c
//complex numbers - using strings

#include<stdio.h>
#include<stdlib.h>
#include<math.h>

float atof1(char *arr){
    int i,j,k;
    float val;
    char c;
    i=0;
    j=0;
    val=0;
    k=0;
    while ((c = *(arr+i))!='\0'){
        if (c!='.'){
            val =(val*10)+(c-'0');
            if (k == 1){
                --j;
            }
        }
        if (c=='.'){ if (k == 1) return 0; k=1;}
        ++i;
    }
    val = val*pow(10,j);
    return val;
}

float RPofCN(char str[])
{
```



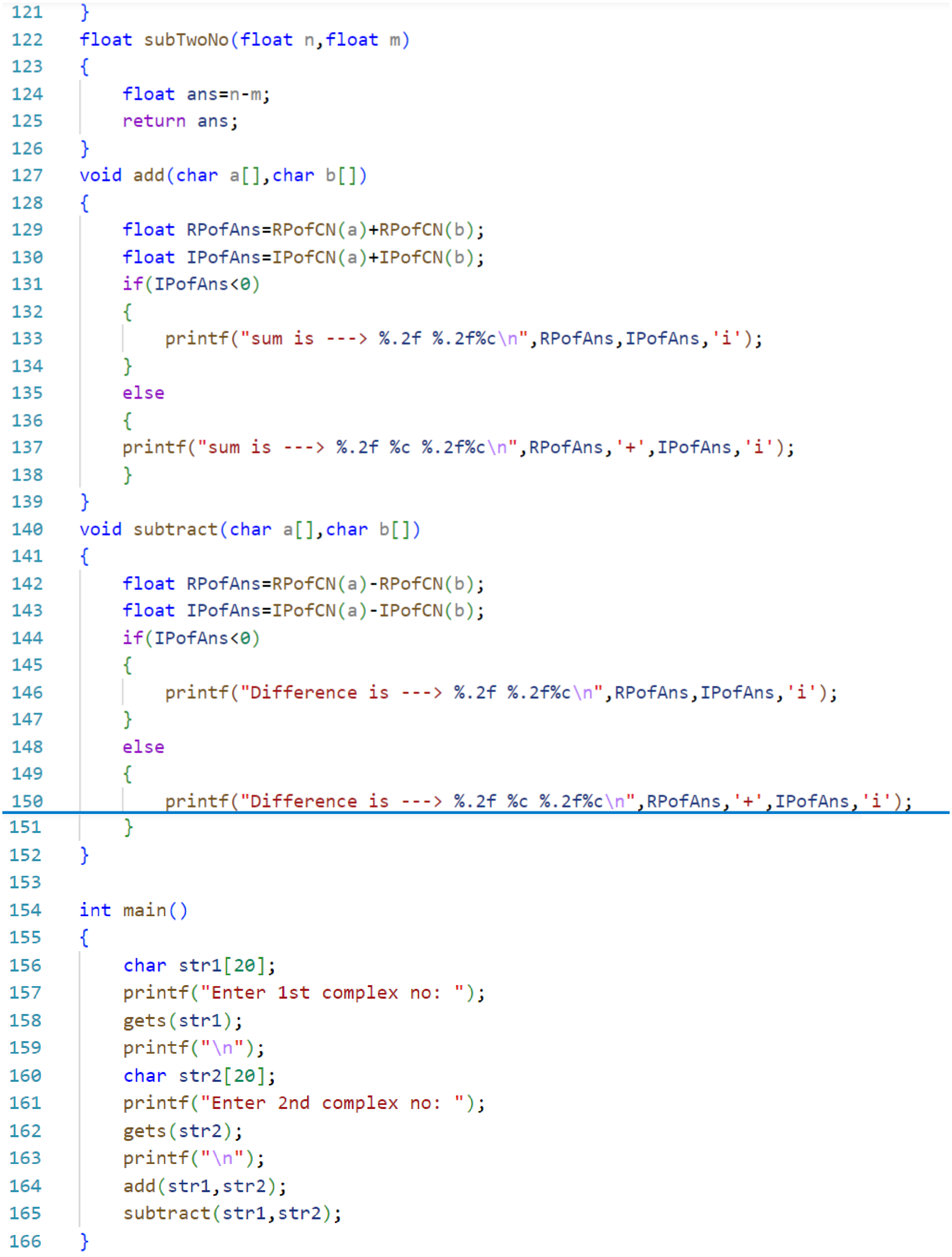
```c
    int i=0,k=0;
    while(str[i]!='\0')
    {
        if(str[i]=='+'||str[i]=='-')
        {
            k=1;
        }
        i++;
    }
    if(k)
    {
        float ans1=atof(str);
        return ans1;
    }
i=0;
    while(str[i]!='\0')
    {
        if(str[i]=='i')
        {
            k=1;
        }
        i++;
    }
    if(k)
    {
        float ans2=0.0;
        return ans2;
    }
    else
    {
```



```
        float ans3=atof(str);
        return ans3;
    }
}
float IPofCN(char str[])
{
    int i=0,si=0,k=0;
    while(str[i]!='\0')
    {
        if(str[i]=='+'||str[i]=='-')
        {
            k=1;
            if(str[i]=='-')
            {
                si=-1;
            }
            break;
        }
        i++;
    }
    i++;
    int j=0;
    char st[10];
    if(k)
    {
        while(str[i]!='\0')
        {
            st[j++]=str[i++];
        }
        float ans1=atof(st);
```

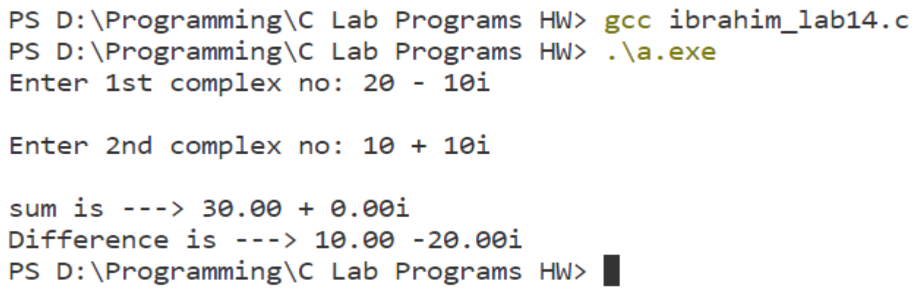


```
        if(si==-1)
        {
            ans1=0.0-ans1;
        }
        return ans1;
    }
    i=0,si=0;
    while(str[i]!='\0')
    {
        if(str[i]=='i')
        {
            k=1;
        }
        i++;
    }
    if(k)
    {
        float ans2=atof(str);
        return ans2;
    }
    else
    {
        float ans3=0.0;
        return ans3;
    }
}
float addTwoNo(float n,float m)
{
    float ans=n+m;
    return ans;
```



```
}
float subTwoNo(float n,float m)
{
    float ans=n-m;
    return ans;
}
void add(char a[],char b[])
{
    float RPofAns=RPofCN(a)+RPofCN(b);
    float IPofAns=IPofCN(a)+IPofCN(b);
    if(IPofAns<0)
    {
        printf("sum is ---> %.2f %.2f%c\n",RPofAns,IPofAns,'i');
    }
    else
    {
    printf("sum is ---> %.2f %c %.2f%c\n",RPofAns,'+',IPofAns,'i');
    }
}
void subtract(char a[],char b[])
{
    float RPofAns=RPofCN(a)-RPofCN(b);
    float IPofAns=IPofCN(a)-IPofCN(b);
    if(IPofAns<0)
    {
        printf("Difference is ---> %.2f %.2f%c\n",RPofAns,IPofAns,'i');
    }
    else
    {
        printf("Difference is ---> %.2f %c %.2f%c\n",RPofAns,'+',IPofAns,'i');
    }
}

int main()
{
    char str1[20];
    printf("Enter 1st complex no: ");
    gets(str1);
    printf("\n");
    char str2[20];
    printf("Enter 2nd complex no: ");
    gets(str2);
    printf("\n");
    add(str1,str2);
    subtract(str1,str2);
}
```

Output: 14



```
PS D:\Programming\C Lab Programs HW> gcc ibrahim_lab14.c
PS D:\Programming\C Lab Programs HW> .\a.exe
Enter 1st complex no: 20 - 10i

Enter 2nd complex no: 10 + 10i

sum is ---> 30.00 + 0.00i
Difference is ---> 10.00 -20.00i
PS D:\Programming\C Lab Programs HW> 
```