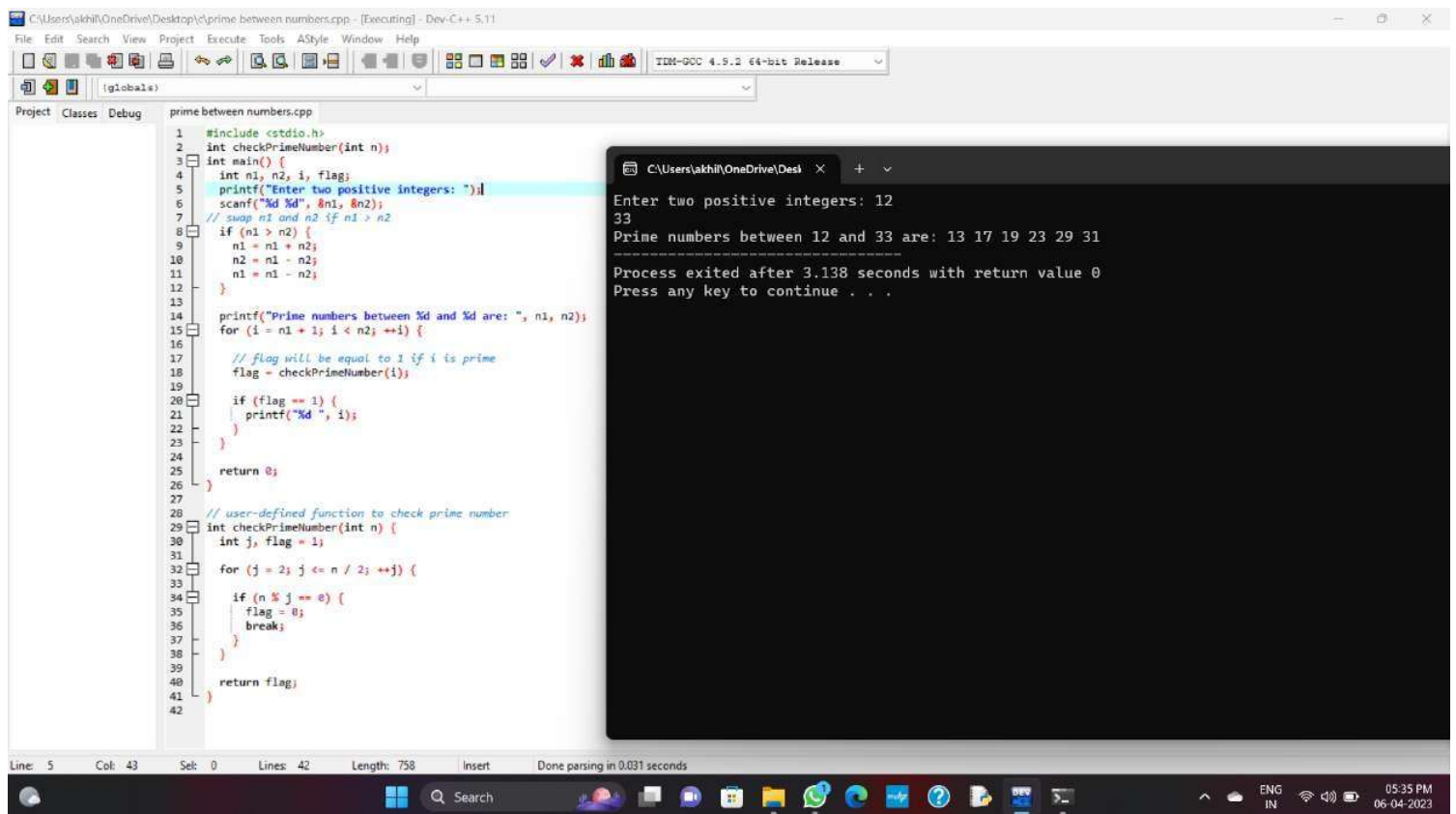


ASSIGNMENT – 3: C PROGRAMMING

COURSE CODE: CSA0268

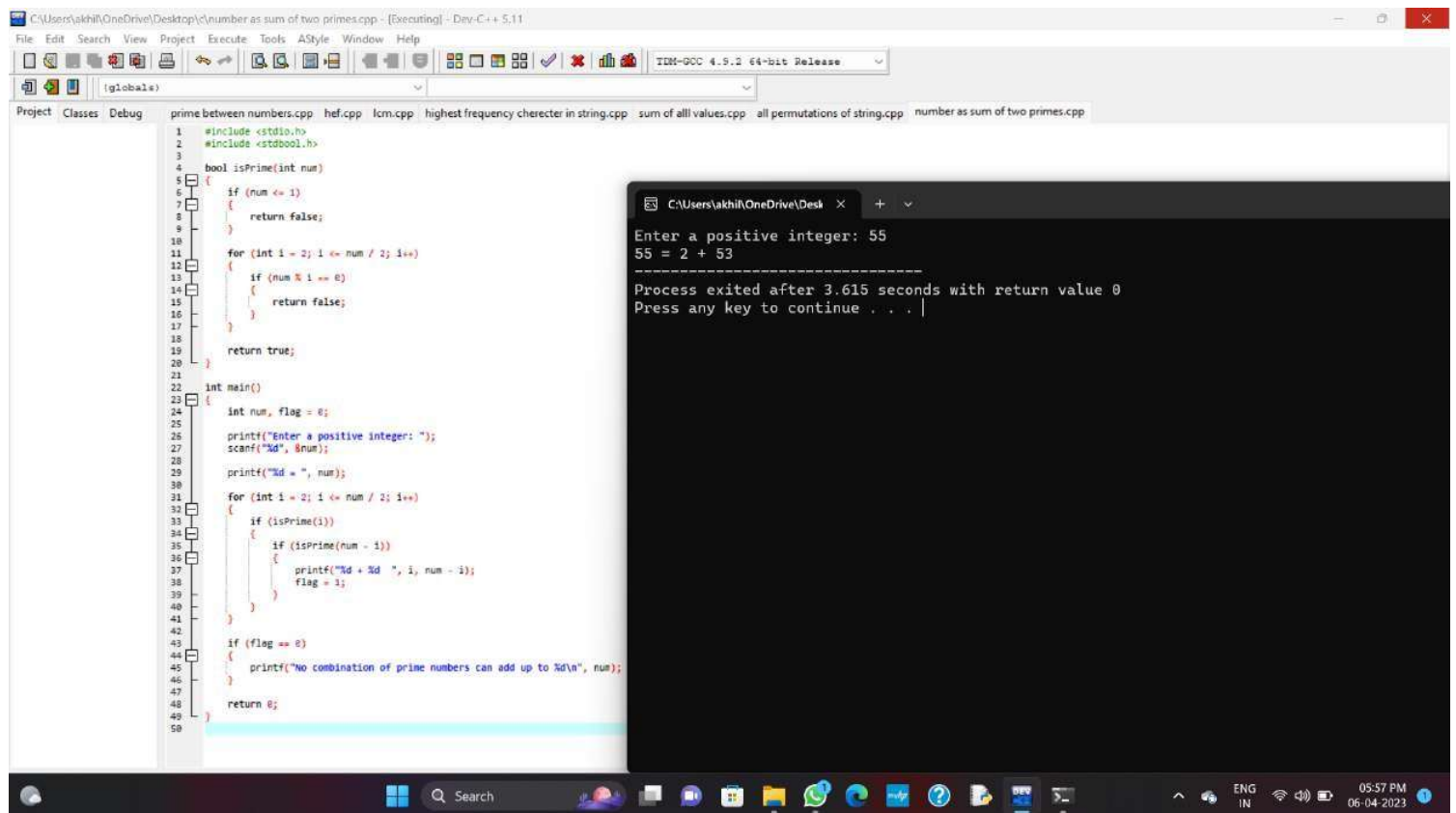
1.C Program to Display Prime Numbers Between Intervals Using Function



```
1 #include <stdio.h>
2 int checkPrimeNumber(int n);
3 int main() {
4     int n1, n2, i, flag;
5     printf("Enter two positive integers: ");
6     scanf("%d %d", &n1, &n2);
7     // swap n1 and n2 if n1 > n2
8     if (n1 > n2) {
9         n1 = n1 + n2;
10        n2 = n1 - n2;
11        n1 = n1 - n2;
12    }
13    printf("Prime numbers between %d and %d are: ", n1, n2);
14    for (i = n1 + 1; i < n2; ++i) {
15        // flag will be equal to 1 if i is prime
16        flag = checkPrimeNumber(i);
17        if (flag == 1) {
18            printf("%d ", i);
19        }
20    }
21    return 0;
22 }
23 // user-defined function to check prime number
24 int checkPrimeNumber(int n) {
25     int j, flag = 1;
26     for (j = 2; j <= n / 2; ++j) {
27         if (n % j == 0) {
28             flag = 0;
29             break;
30         }
31     }
32     return flag;
33 }
```

Enter two positive integers: 12 33
Prime numbers between 12 and 33 are: 13 17 19 23 29 31
Process exited after 3.138 seconds with return value 0
Press any key to continue . . .

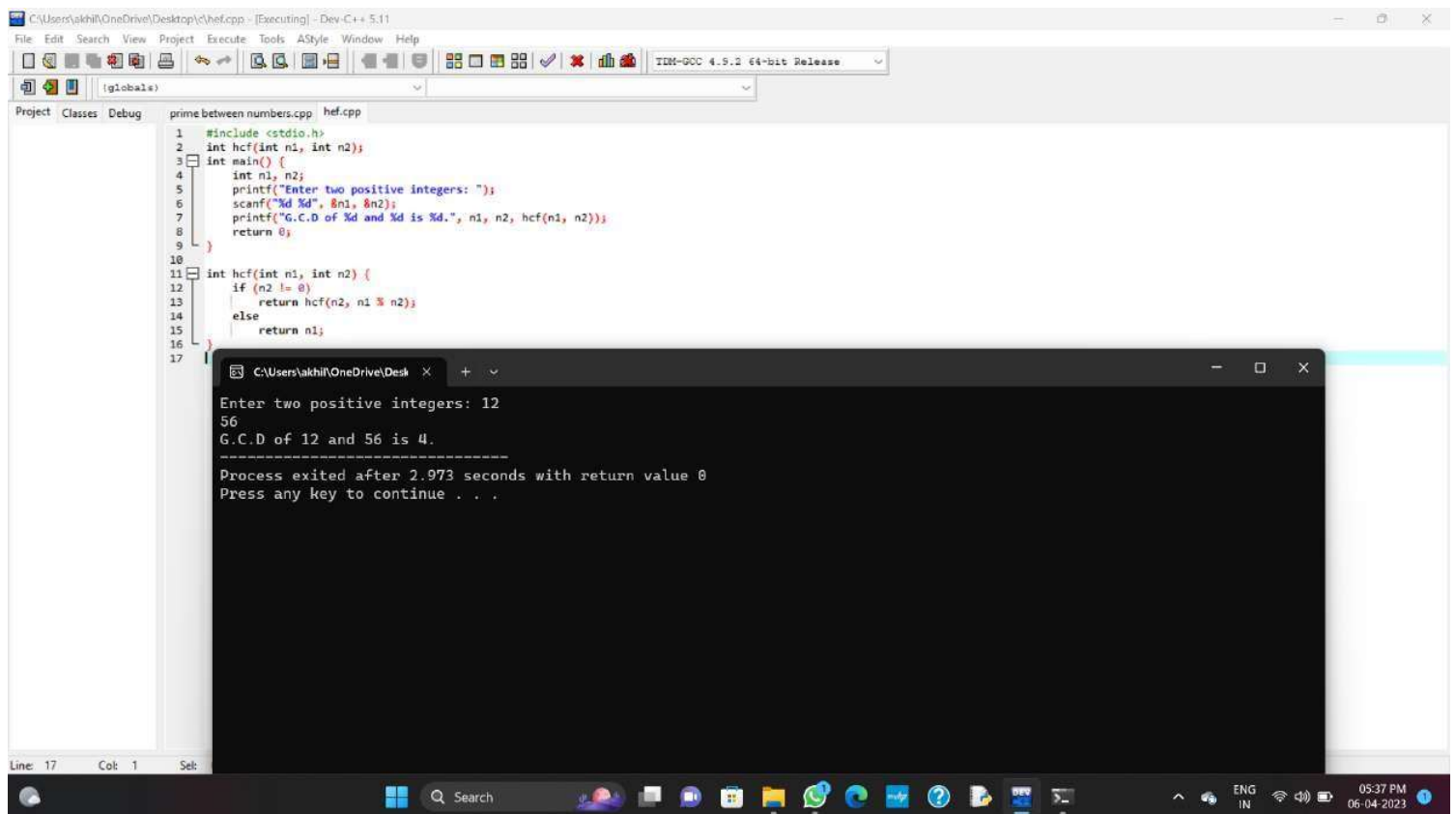
2.C Program to Check Whether a Number can be Expressed as Sum of Two Prime Numbers



```
1 #include <stdio.h>
2 #include <stdbool.h>
3
4 bool isPrime(int num)
5 {
6     if (num <= 1)
7     {
8         return false;
9     }
10
11     for (int i = 2; i <= num / 2; i++)
12     {
13         if (num % i == 0)
14         {
15             return false;
16         }
17     }
18
19     return true;
20 }
21
22 int main()
23 {
24     int num, flag = 0;
25
26     printf("Enter a positive integer: ");
27     scanf("%d", &num);
28
29     printf("%d = ", num);
30
31     for (int i = 2; i <= num / 2; i++)
32     {
33         if (isPrime(i))
34         {
35             if (isPrime(num - i))
36             {
37                 printf("%d + %d ", i, num - i);
38                 flag = 1;
39             }
40         }
41     }
42
43     if (flag == 0)
44     {
45         printf("No combination of prime numbers can add up to %d\n", num);
46     }
47
48     return 0;
49 }
```

```
Enter a positive integer: 55
55 = 2 + 53
-----
Process exited after 3.615 seconds with return value 0
Press any key to continue . . .
```

3.C Program to Find GCD of Two Numbers using Recursion



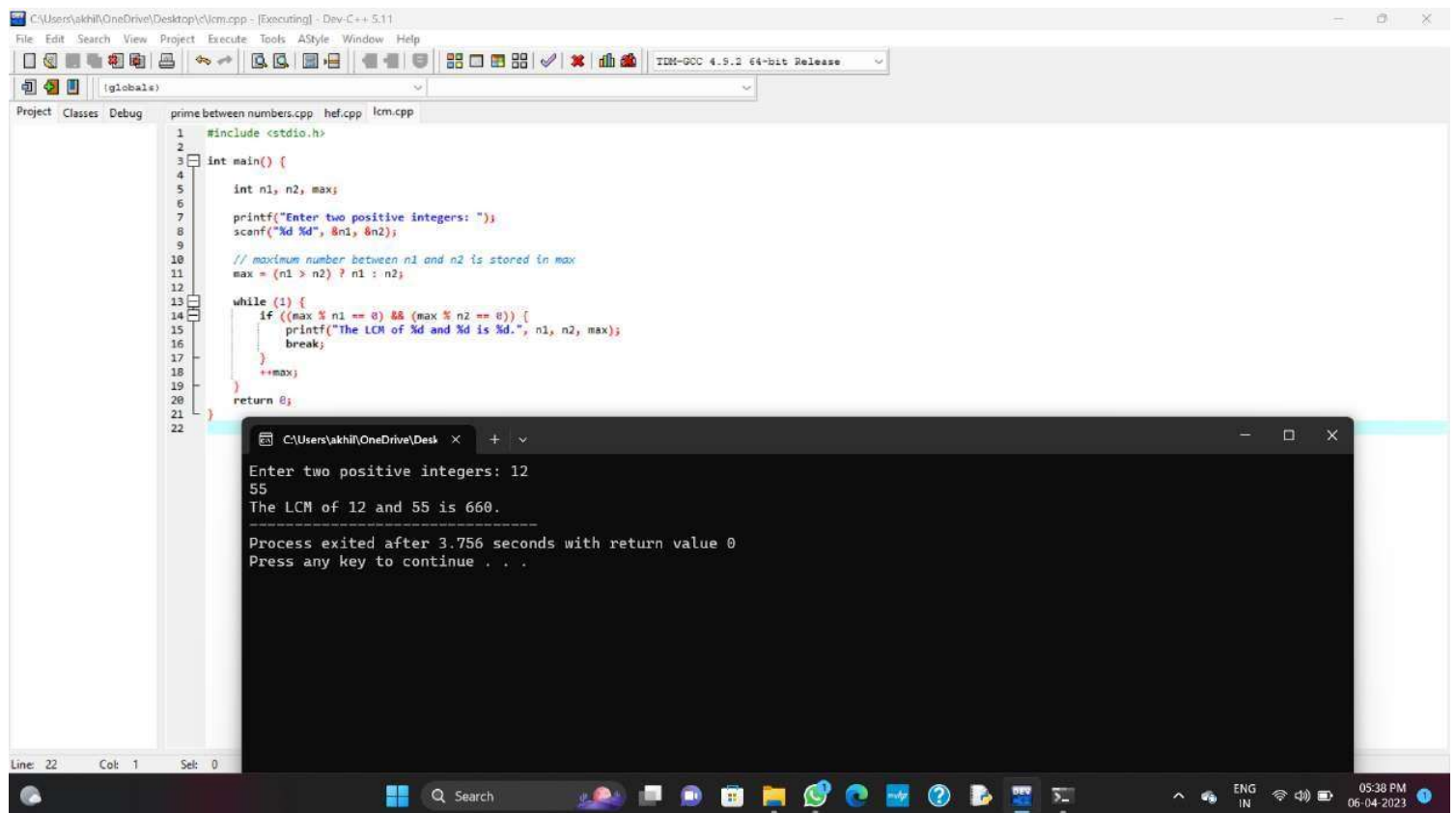
The screenshot displays a C++ IDE with a project named 'prime between numbers.cpp' and a file named 'hcf.cpp'. The code in 'hcf.cpp' implements a recursive function to find the GCD of two numbers. The main function prompts the user to enter two positive integers, reads them, and prints the GCD using the recursive function. The execution output shows the user entering 12 and 56, resulting in a GCD of 4. The process exited after 2.973 seconds with a return value of 0.

```
1 #include <stdio.h>
2 int hcf(int n1, int n2);
3 int main() {
4     int n1, n2;
5     printf("Enter two positive integers: ");
6     scanf("%d %d", &n1, &n2);
7     printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));
8     return 0;
9 }
10
11 int hcf(int n1, int n2) {
12     if (n2 != 0)
13         return hcf(n2, n1 % n2);
14     else
15         return n1;
16 }
17
```

Enter two positive integers: 12
56
G.C.D of 12 and 56 is 4.

Process exited after 2.973 seconds with return value 0
Press any key to continue . . .

4. C Program to Find LCM of Two Numbers



The screenshot shows a Dev-C++ IDE with a C program to find the LCM of two numbers. The program is named `lcm.cpp` and is located at `C:\Users\akshil\OneDrive\Desktop\lcm.cpp`. The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int n1, n2, max;
5
6     printf("Enter two positive integers: ");
7     scanf("%d %d", &n1, &n2);
8
9     // maximum number between n1 and n2 is stored in max
10    max = (n1 > n2) ? n1 : n2;
11
12    while (1) {
13        if ((max % n1 == 0) && (max % n2 == 0)) {
14            printf("The LCM of %d and %d is %d.", n1, n2, max);
15            break;
16        }
17        ++max;
18    }
19    return 0;
20 }
```

The program is executed, and the output is shown in the console window:

```
Enter two positive integers: 12
55
The LCM of 12 and 55 is 660.
Process exited after 3.756 seconds with return value 0
Press any key to continue . . .
```

The status bar at the bottom indicates the current line is 22, column is 1, and selection is 0. The system tray shows the date and time as 05:38 PM on 06-04-2023.

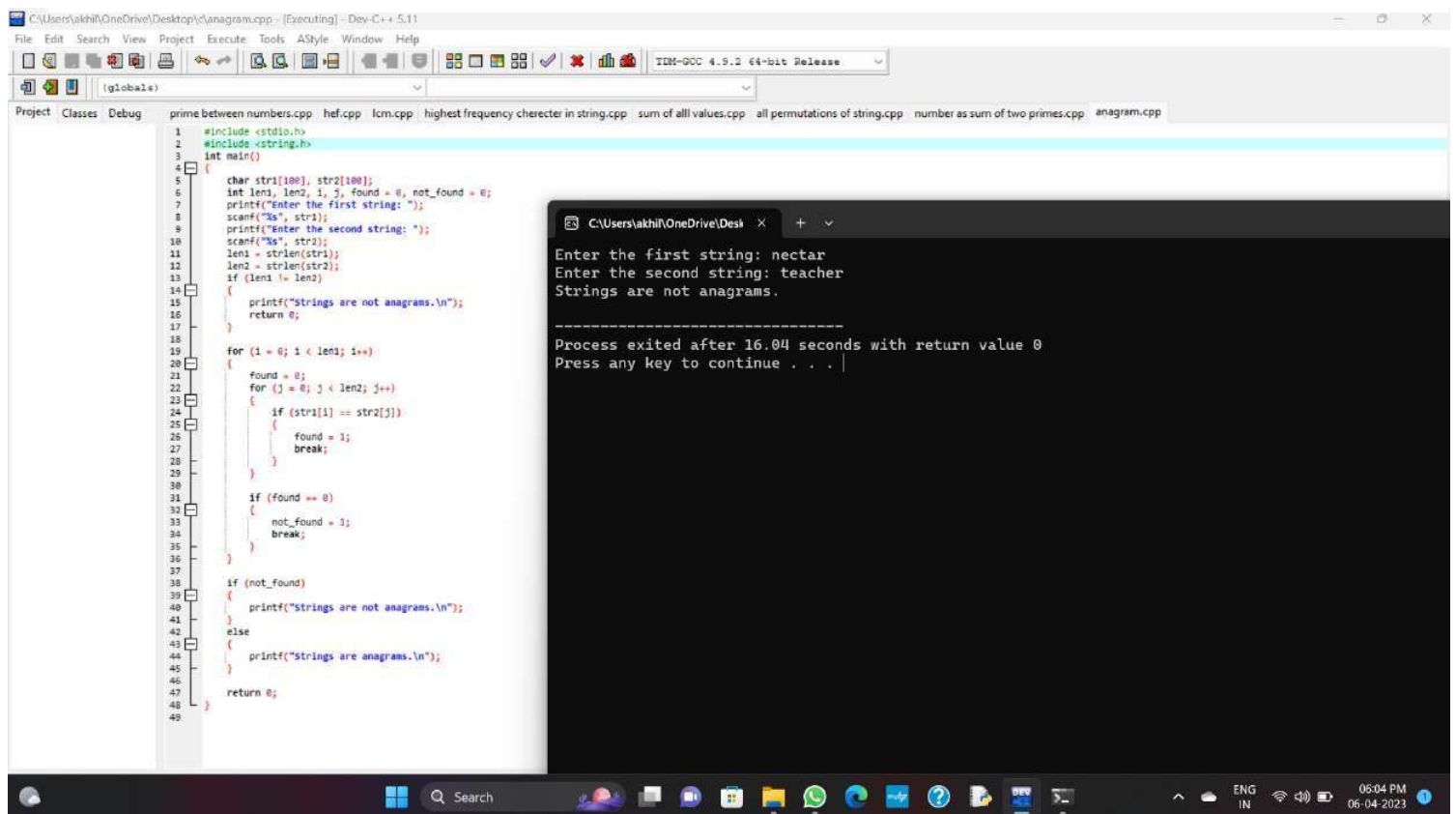
5.C Program to Find Highest Frequency Character in a String

```
1 #include <stdio.h>
2 #include <string.h>
3
4 #define MAX_SIZE 100
5
6 int main()
7 {
8     char str[MAX_SIZE];
9     int i, j, max_freq, freq;
10    char max_char;
11
12    printf("Enter a string : ");
13    fgets(str, sizeof(str), stdin);
14
15    max_freq = -1;
16    for(i = 0; str[i] != '\0'; i++)
17    {
18        freq = 1;
19        for(j = i+1; str[j] != '\0'; j++)
20        {
21            if(str[j] == str[i])
22            {
23                freq++;
24            }
25        }
26
27        if(freq > max_freq)
28        {
29            max_freq = freq;
30            max_char = str[i];
31        }
32    }
33
34    printf("Max repeated character in the string = %c\n", max_char);
35    printf("It occurs %d times\n", max_freq);
36
37    return 0;
38 }
```

Enter a string : ghjklgfd
Max repeated character in the string = g
It occurs 2 times

Process exited after 9.018 seconds with return value 0
Press any key to continue . . .

6. **Anagram Program in C:** Two strings are said to be **anagrams** if they satisfy two conditions, the length of both strings must be equal to each other and second the strings must have the same set of characters.



The screenshot displays a C++ IDE with the file `anagram.cpp` open. The code implements a function to check if two strings are anagrams. It includes `<stdio.h>` and `<string.h>`. In the `main` function, it declares two character arrays `str1` and `str2` of size 100, and variables for lengths `len1`, `len2`, and flags `found` and `not_found`. It prompts the user to enter two strings. If the lengths are not equal, it prints "Strings are not anagrams." and returns 0. If lengths are equal, it uses nested loops to compare characters. If a mismatch is found, it sets `not_found = 1` and breaks. If no mismatch is found after the loops, it prints "Strings are anagrams." and returns 0.

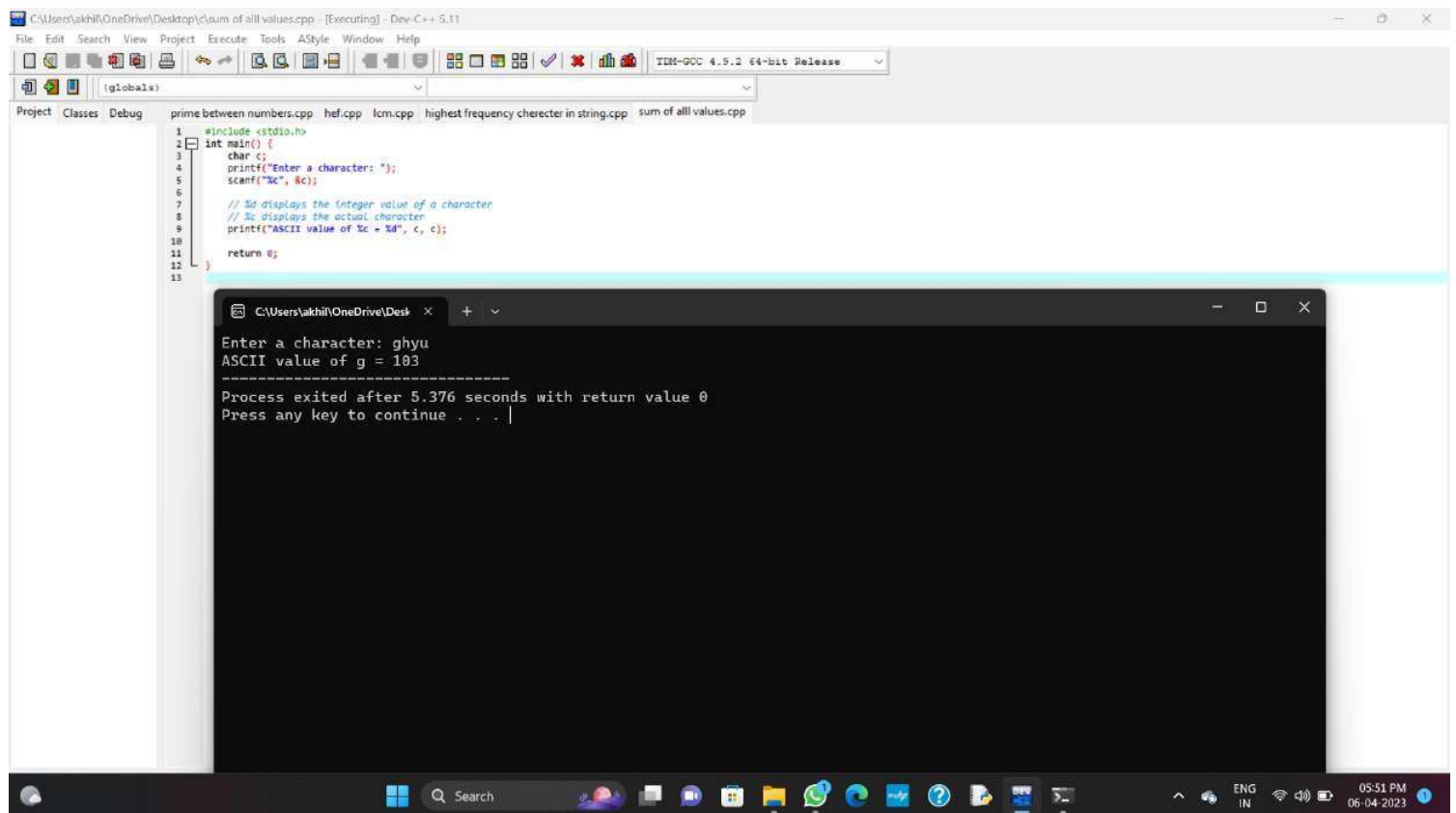
```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char str1[100], str2[100];
6     int len1, len2, i, j, found = 0, not_found = 0;
7     printf("Enter the first string: ");
8     scanf("%s", str1);
9     printf("Enter the second string: ");
10    scanf("%s", str2);
11    len1 = strlen(str1);
12    len2 = strlen(str2);
13    if (len1 != len2)
14    {
15        printf("Strings are not anagrams.\n");
16        return 0;
17    }
18    for (i = 0; i < len1; i++)
19    {
20        found = 0;
21        for (j = 0; j < len2; j++)
22        {
23            if (str1[i] == str2[j])
24            {
25                found = 1;
26                break;
27            }
28        }
29        if (found == 0)
30        {
31            not_found = 1;
32            break;
33        }
34    }
35    if (not_found)
36    {
37        printf("Strings are not anagrams.\n");
38    }
39    else
40    {
41        printf("Strings are anagrams.\n");
42    }
43    return 0;
44 }
```

The execution output shows the program running with the inputs "nectar" and "teacher". It correctly identifies them as not anagrams and exits after 16.04 seconds.

```
Enter the first string: nectar
Enter the second string: teacher
Strings are not anagrams.

-----
Process exited after 16.04 seconds with return value 0
Press any key to continue . . .
```

7. C Program to Find the Sum of ASCII Value of All Characters in the String



The screenshot displays a C program in the Dev-C++ IDE, titled "sum of all values.cpp". The program is designed to calculate the sum of the ASCII values of all characters in a string. The code is as follows:

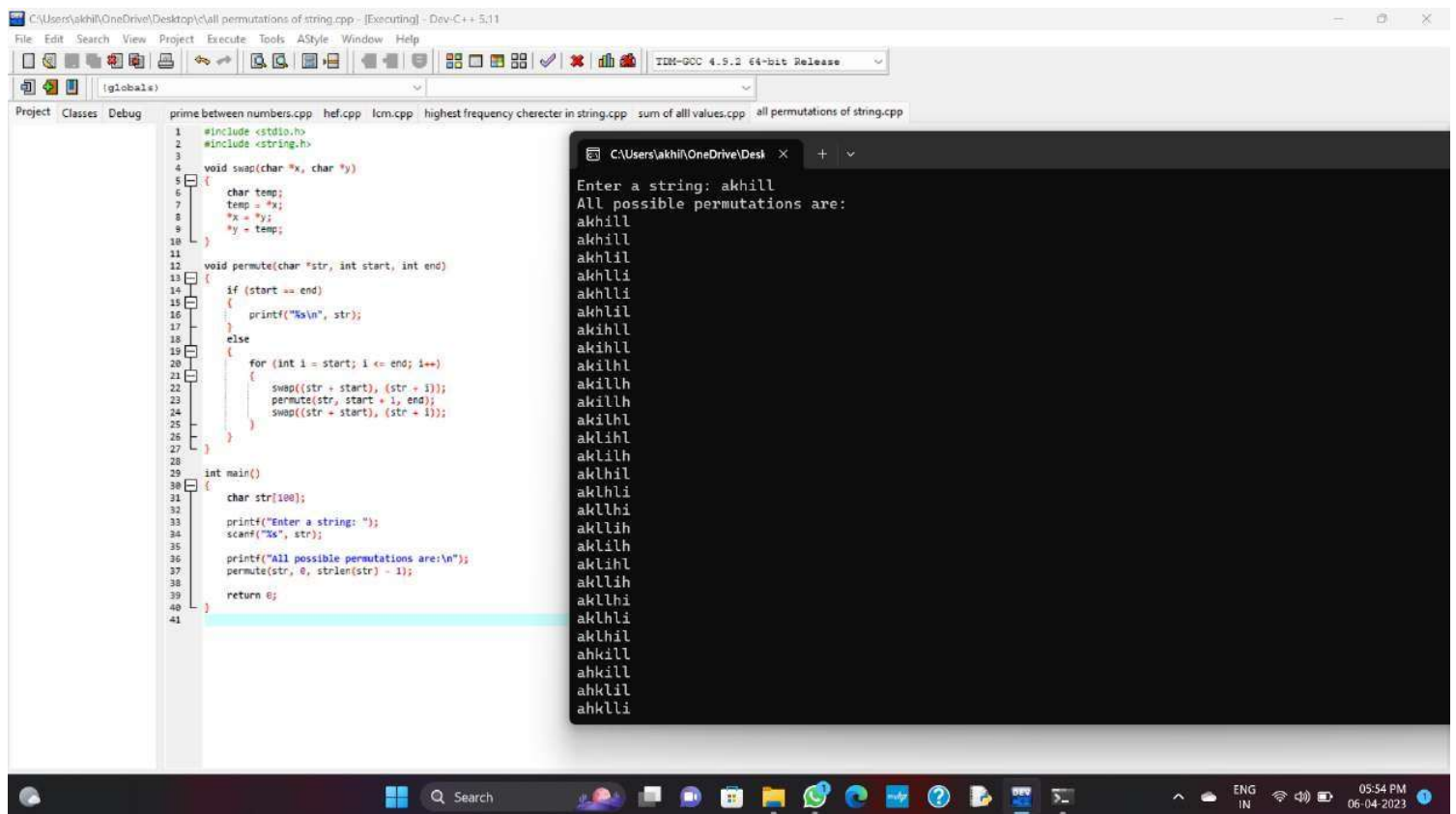
```
1 #include <stdio.h>
2 int main() {
3     char c;
4     printf("Enter a character: ");
5     scanf("%c", &c);
6
7     // %d displays the integer value of a character
8     // %c displays the actual character
9     printf("ASCII value of %c = %d", c, c);
10
11     return 0;
12 }
13
```

The program's output, shown in a separate window, is as follows:

```
Enter a character: ghyu
ASCII value of g = 103
-----
Process exited after 5.376 seconds with return value 0
Press any key to continue . . .
```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a toolbar, and a project explorer on the left. The status bar at the bottom indicates the system time as 05:51 PM on 06-04-2023.

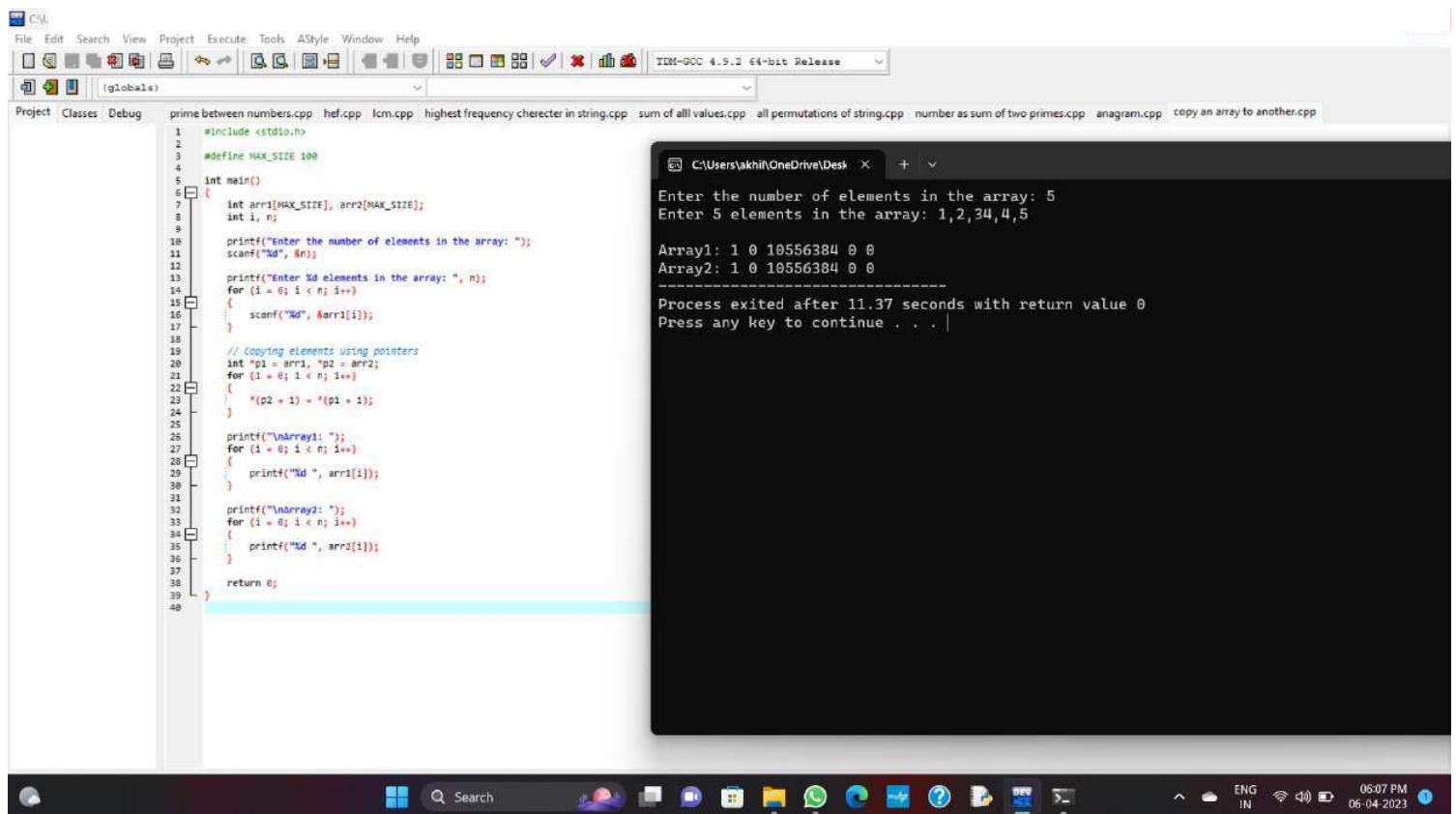
8.C Program to Print All Permutations of a Given String



```
1 #include <stdio.h>
2 #include <string.h>
3
4 void swap(char *x, char *y)
5 {
6     char temp;
7     temp = *x;
8     *x = *y;
9     *y = temp;
10 }
11
12 void permute(char *str, int start, int end)
13 {
14     if (start == end)
15     {
16         printf("%s\n", str);
17     }
18     else
19     {
20         for (int i = start; i <= end; i++)
21         {
22             swap((str + start), (str + i));
23             permute(str, start + 1, end);
24             swap((str + start), (str + i));
25         }
26     }
27 }
28
29 int main()
30 {
31     char str[100];
32     printf("Enter a string: ");
33     scanf("%s", str);
34
35     printf("All possible permutations are:\n");
36     permute(str, 0, strlen(str) - 1);
37
38     return 0;
39 }
40
41
```

```
Enter a string: akhill
All possible permutations are:
akhill
akhill
akhlil
akhlil
akhlili
akhlili
akihll
akihll
akihlil
akihlil
akihlli
akihlli
akilhl
akilhl
akilhll
akilhll
akilhli
akilhli
aklhlil
aklhlil
aklhlih
aklhlih
aklhlli
aklhlli
akllhi
akllhi
akllhli
akllhli
akllhil
akllhil
aklllil
aklllil
aklllli
aklllli
```

9. Write a C program to copy one array elements to another array using pointers. How to copy array elements from one array to another array using pointers. Logic to copy one array to another array using pointers in C programming.



```
1 #include <stdio.h>
2
3 #define MAX_SIZE 100
4
5 int main()
6 {
7     int arr1[MAX_SIZE], arr2[MAX_SIZE];
8     int i, n;
9
10    printf("Enter the number of elements in the array: ");
11    scanf("%d", &n);
12
13    printf("Enter %d elements in the array: ", n);
14    for (i = 0; i < n; i++)
15    {
16        scanf("%d", &arr1[i]);
17    }
18
19    // Copying elements using pointers
20    int *p1 = arr1, *p2 = arr2;
21    for (i = 0; i < n; i++)
22    {
23        *(p2 + i) = *(p1 + i);
24    }
25
26    printf("\nArray1: ");
27    for (i = 0; i < n; i++)
28    {
29        printf("%d ", arr1[i]);
30    }
31
32    printf("\nArray2: ");
33    for (i = 0; i < n; i++)
34    {
35        printf("%d ", arr2[i]);
36    }
37
38    return 0;
39 }
```

Output:

```
Enter the number of elements in the array: 5
Enter 5 elements in the array: 1,2,3,4,5

Array1: 1 0 10556384 0 0
Array2: 1 0 10556384 0 0

Process exited after 11.37 seconds with return value 0
Press any key to continue . . .
```

10. Write a C Program to reverse string using pointers and function.

```
1 #include <stdio.h>
2 #include <string.h>
3
4 void reverse_string(char*);
5
6 int main()
7 {
8     char str[100];
9     printf("Enter any string: ");
10    scanf("%s", str);
11
12    printf("Original string: %s\n", str);
13
14    // calling the function to reverse the string
15    reverse_string(str);
16
17    printf("Reversed string: %s\n", str);
18
19    return 0;
20 }
21
22 void reverse_string(char *str)
23 {
24     int i, j;
25     char temp;
26
27     // swapping characters using pointers
28     for (i = 0, j = strlen(str) - 1; i < j; i++, j--)
29     {
30         temp = *(str + i);
31         *(str + i) = *(str + j);
32         *(str + j) = temp;
33     }
34 }
35
36
```

Enter any string: saveetha
Original string: saveetha
Reversed string: ahteevas

Process exited after 7.122 seconds with return value 0
Press any key to continue . . .