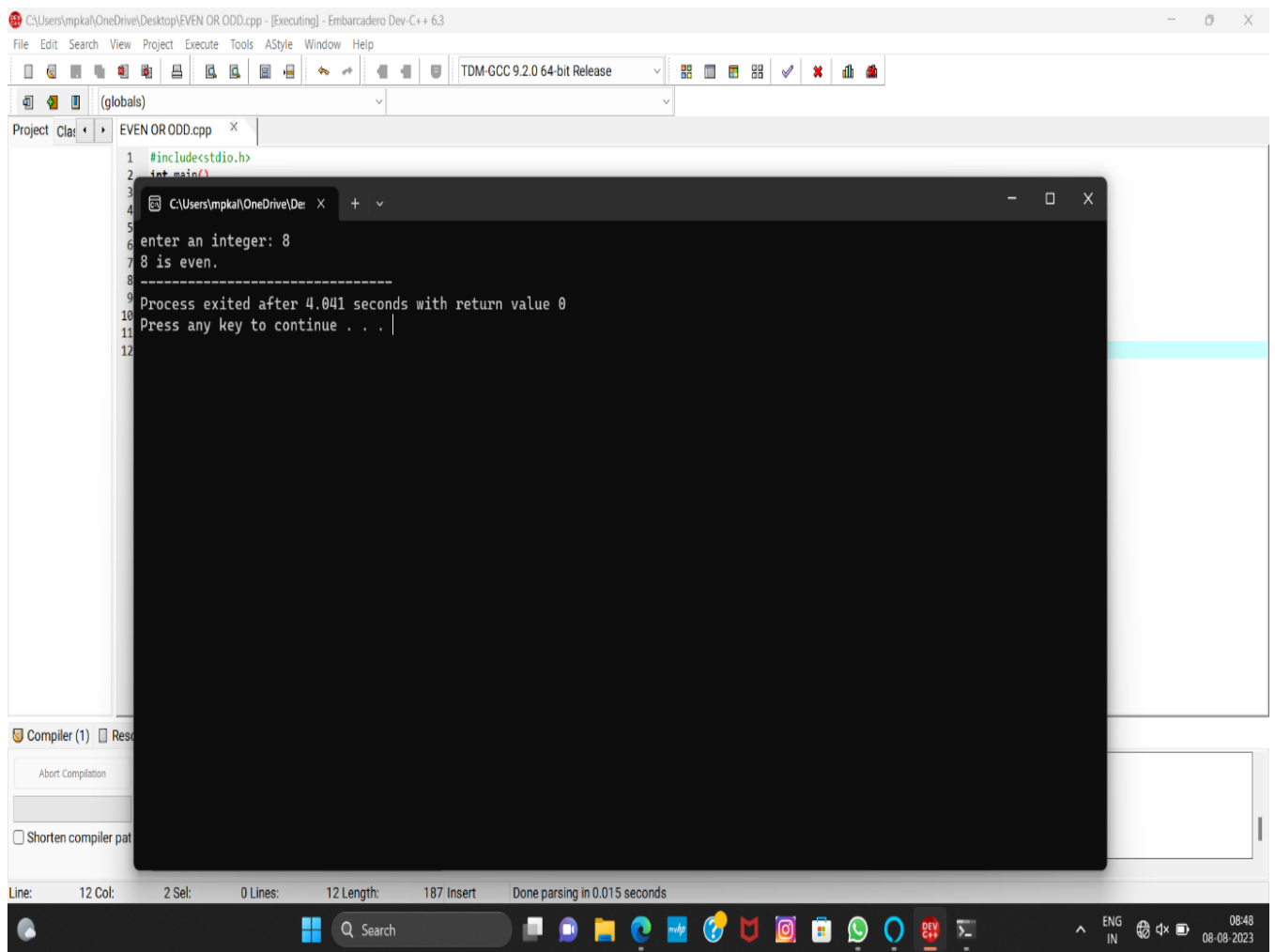


## 1.EVEN OR ODD



## 2.sum of n numbers using for loop

The screenshot shows an IDE window titled "C:\Users\mpkal\OneDrive\Desktop\sum n numbers usin for loop.cpp - [Executing] - Embarcadero Dev-C++ 6.3". The code in the editor is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,i,sum=0;
5     printf("enter the positive number: ");
6     scanf("%d",&n);
7     for(i=0;i<=n;i++)
8     {
9         sum+=1;
10    }
11    printf("sum = %d",sum);
12    return 0;
13 }
```

The terminal window on the right shows the program's execution:

```
C:\Users\mpkal\OneDrive\De: x + v
enter the positive number: 7
sum = 8
-----
Process exited after 13.13 seconds with return value 0
Press any key to continue . . .
```

At the bottom, the compiler output shows 0 errors and 0 warnings. The output filename is "C:\Users\mpkal\OneDrive\Desktop\sum n numbers usin for loop.exe", the output size is "322.8154296875 KiB", and the compilation time is "0.30s".

### 3.sum of even num using while loop

The screenshot displays the Embarcadero Dev-C++ IDE. The main editor window shows a C++ program that calculates the sum of even numbers from 2 to a user-defined value 'n'. The code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,sum=0,i=2;
5     printf("enter the value of n: ");
6     scanf("%d",&n);
7     while(i<=n) {
8         sum+=i;
9         i+=2;
10    }
11    printf("sum of even numbers from 2 to %d is: %d\n",n,sum);
12    return 0;
13 }
```

The 'even number using while loop.cpp' file is selected in the Project Explorer. The Compiler window at the bottom shows successful compilation with 0 errors and 0 warnings. The output file is 'even number using while loop.exe'.

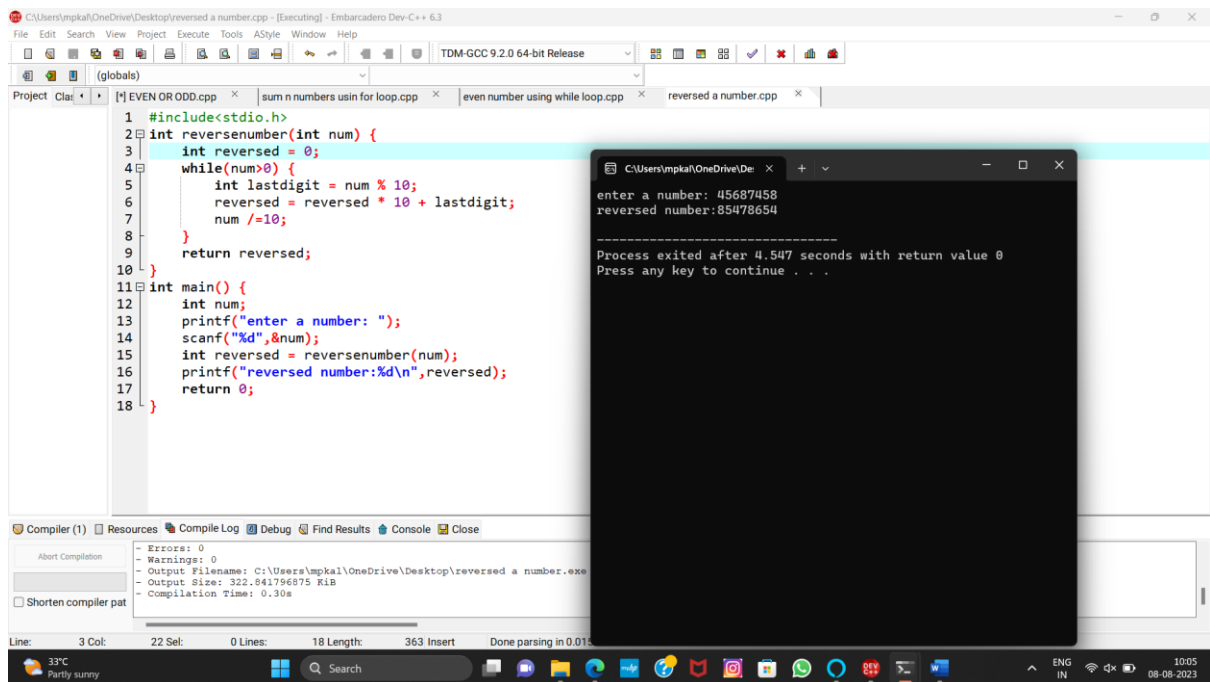
An external terminal window is overlaid on the IDE, showing the program's execution:

```
enter the value of n: 6
sum of even numbers from 2 to 6 is: 12

-----
Process exited after 25.82 seconds with return value 0
Press any key to continue . . . |
```

The Windows taskbar at the bottom indicates the system time is 09:44 on 08-08-2023, with a temperature of 31°C and 'Partly sunny' weather.

## 4.reversed a number



The image shows a screenshot of an IDE (Embarcadero Dev-C++) with a C++ project named "reversed a number.cpp". The code implements a function to reverse a number using a while loop. The main function prompts the user to enter a number, calls the reversal function, and prints the result.

```
1 #include<stdio.h>
2 int reversenumber(int num) {
3     int reversed = 0;
4     while(num>0) {
5         int lastdigit = num % 10;
6         reversed = reversed * 10 + lastdigit;
7         num /=10;
8     }
9     return reversed;
10 }
11 int main() {
12     int num;
13     printf("enter a number: ");
14     scanf("%d",&num);
15     int reversed = reversenumber(num);
16     printf("reversed number:%d\n",reversed);
17     return 0;
18 }
```

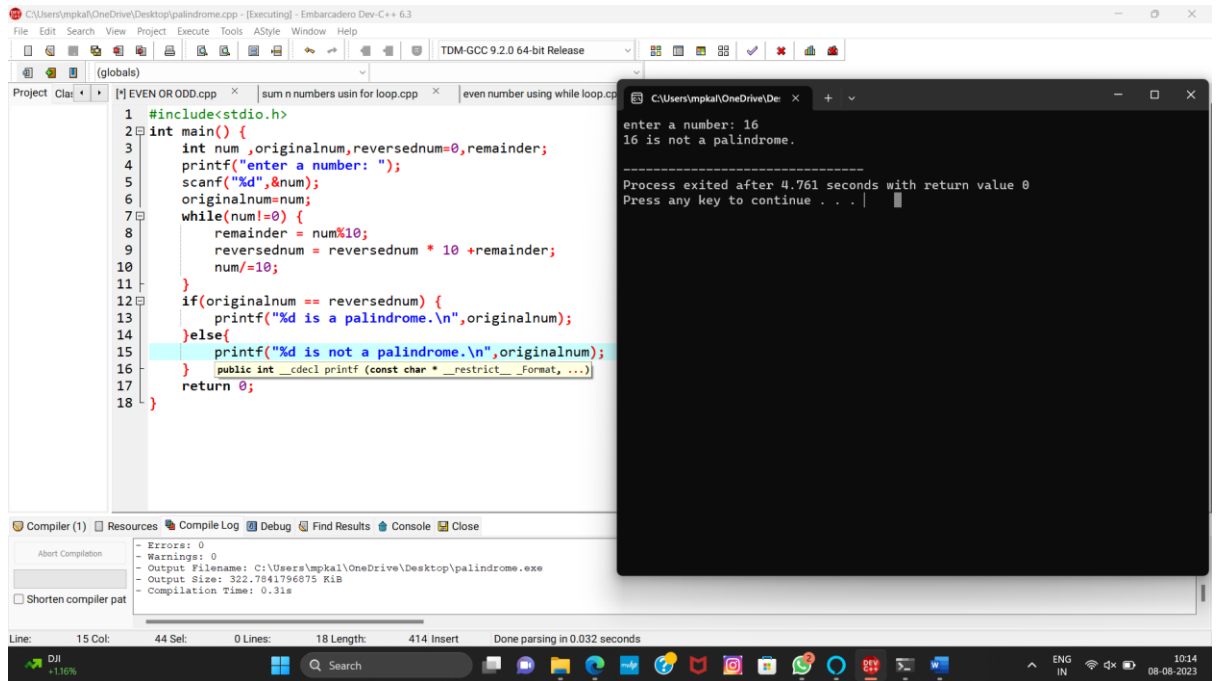
The output window shows the execution results:

```
enter a number: 45687458
reversed number:85478654

-----
Process exited after 4.547 seconds with return value 0
Press any key to continue . . .
```

The compiler window shows no errors or warnings, and the output filename is "C:\Users\mpkal\OneDrive\Desktop\reversed a number.exe".

## 5.palindrome



The screenshot displays a C++ IDE with a project named "palindrome.cpp". The code is as follows:

```
1 #include<stdio.h>
2 int main() {
3     int num ,originalnum,reversednum=0,remainder;
4     printf("enter a number: ");
5     scanf("%d",&num);
6     originalnum=num;
7     while(num!=0) {
8         remainder = num%10;
9         reversednum = reversednum * 10 +remainder;
10        num/=10;
11    }
12    if(originalnum == reversednum) {
13        printf("%d is a palindrome.\n",originalnum);
14    }else{
15        printf("%d is not a palindrome.\n",originalnum);
16    }
17    return 0;
18 }
```

The IDE's console window shows the program's execution with the input "16". The output indicates that "16 is not a palindrome." and the process exited after 4.761 seconds with a return value of 0.

Compiler (1) Resources Compile Log Debug Find Results Console Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\mpkal\OneDrive\Desktop\palindrome.exe
- Output Size: 322.7841796875 KiB
- Compilation Time: 0.31s

Shorten compiler pat

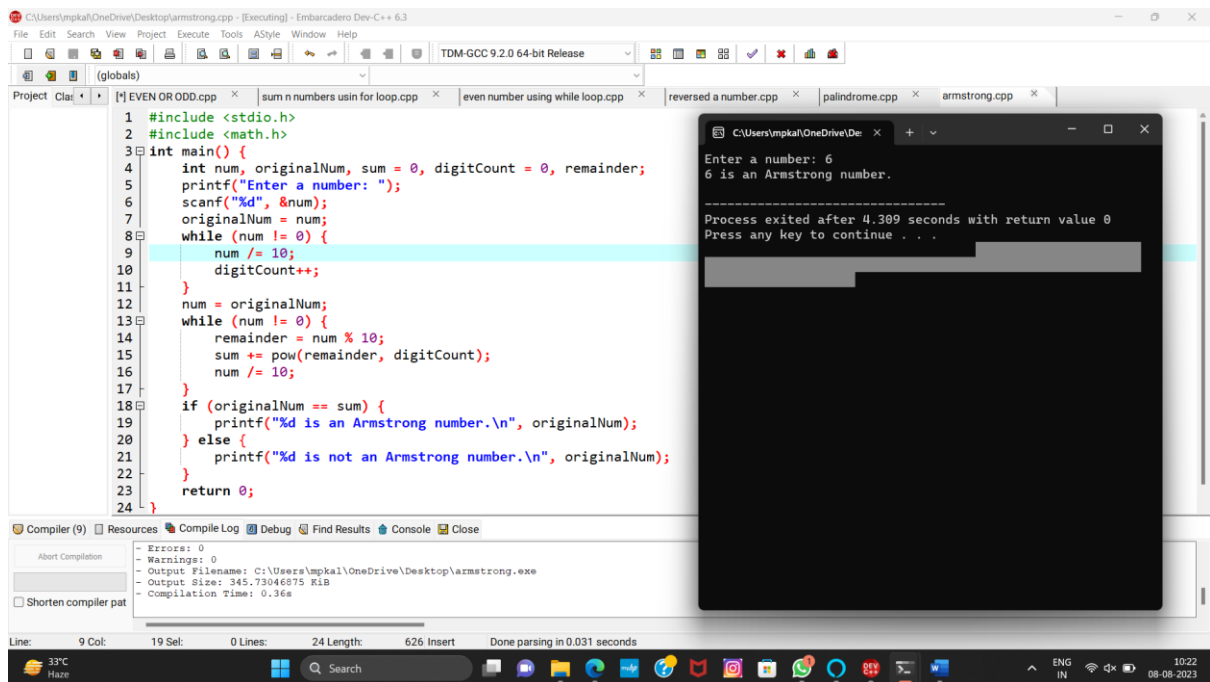
Line: 15 Col: 44 Sel: 0 Lines: 18 Length: 414 Insert Done parsing in 0.032 seconds

DJI +1.6%

Search

ENG IN 10:14 08-08-2023

## 6.armstrong number



The screenshot displays a C++ program in an IDE, specifically using the TDM-GCC 9.2.0 64-bit Release compiler. The code is designed to check if a given number is an Armstrong number. It prompts the user to enter a number, calculates the sum of the cubes of its digits, and compares it to the original number. The output window shows the program's execution for the input '6', confirming it is an Armstrong number.

```
1 #include <stdio.h>
2 #include <math.h>
3 int main() {
4     int num, originalNum, sum = 0, digitCount = 0, remainder;
5     printf("Enter a number: ");
6     scanf("%d", &num);
7     originalNum = num;
8     while (num != 0) {
9         num /= 10;
10        digitCount++;
11    }
12    num = originalNum;
13    while (num != 0) {
14        remainder = num % 10;
15        sum += pow(remainder, digitCount);
16        num /= 10;
17    }
18    if (originalNum == sum) {
19        printf("%d is an Armstrong number.\n", originalNum);
20    } else {
21        printf("%d is not an Armstrong number.\n", originalNum);
22    }
23    return 0;
24 }
```

Compiler (9) | Resources | Compile Log | Debug | Find Results | Console | Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\mpkal\OneDrive\Desktop\armstrong.exe
- Output Size: 345,730,468,75 KiB
- Compilation Time: 0.36s

Shorten compiler path

Line: 9 Col: 19 Sel: 0 Lines: 24 Length: 626 Insert Done parsing in 0.031 seconds

33°C  
Haze

Search

ENG  
IN

10:22  
08-08-2023

Enter a number: 6  
6 is an Armstrong number.  
-----  
Process exited after 4.389 seconds with return value 0  
Press any key to continue . . .

## 7.factorial num using without recursion

The image shows a screenshot of a C++ IDE (Embarcadero Dev-C++ 6.3) with a project named "palindrome.cpp". The code in the editor calculates the factorial of a number using a loop. The output window shows the program's execution for the input 5, resulting in the factorial 120.

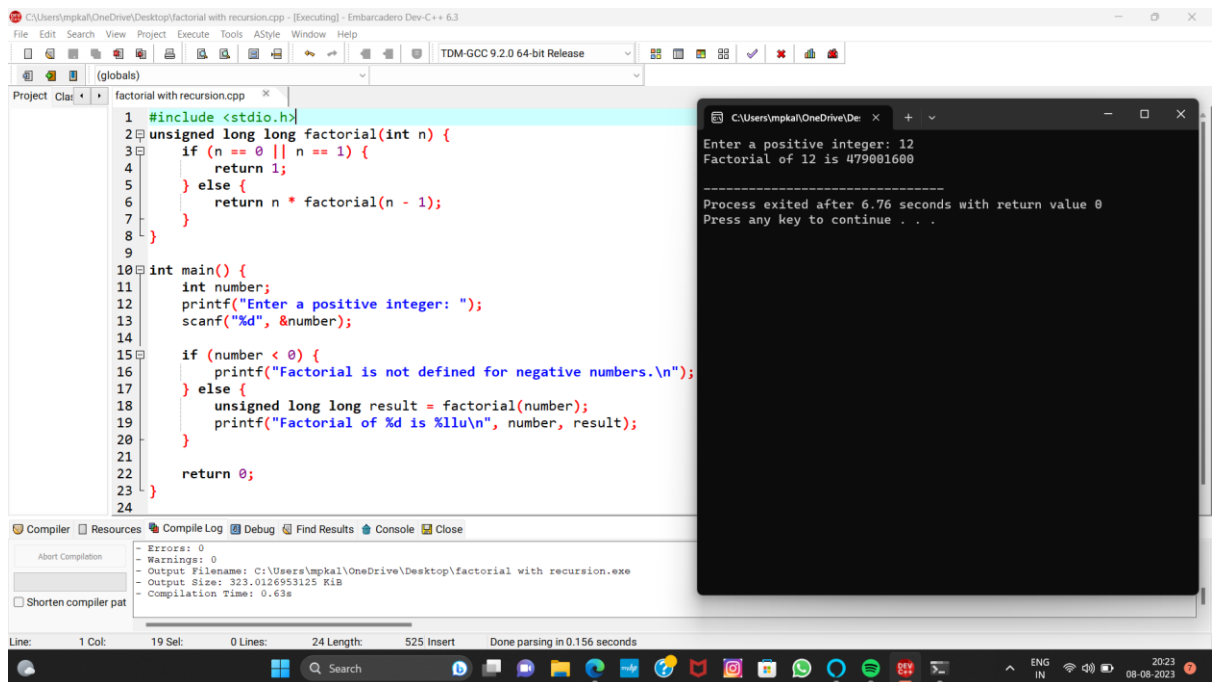
```
1 #include <stdio.h>
2
3 int main() {
4     int num;
5     unsigned long long factorial = 1;
6
7     printf("Enter a number: ");
8     scanf("%d", &num);
9
10    if (num < 0) {
11        printf("Factorial is not defined for negative numbers.\n");
12    } else {
13        for (int i = 1; i <= num; i++) {
14            factorial *= i;
15        }
16        printf("Factorial of %d is %llu\n", num, factorial);
17    }
18    return 0;
19 }
```

Output:

```
Enter a number: 5
Factorial of 5 is 120

-----
Process exited after 1.69 seconds with return value 0
Press any key to continue . . .
```

## 8.factorial with recursion



```
1 #include <stdio.h>
2 unsigned long long factorial(int n) {
3     if (n == 0 || n == 1) {
4         return 1;
5     } else {
6         return n * factorial(n - 1);
7     }
8 }
9
10 int main() {
11     int number;
12     printf("Enter a positive integer: ");
13     scanf("%d", &number);
14
15     if (number < 0) {
16         printf("Factorial is not defined for negative numbers.\n");
17     } else {
18         unsigned long long result = factorial(number);
19         printf("Factorial of %d is %llu\n", number, result);
20     }
21
22     return 0;
23 }
24
```

Compiler: TDM-GCC 9.2.0 64-bit Release

Console:

```
Enter a positive integer: 12
Factorial of 12 is 479001600

Process exited after 6.76 seconds with return value 0
Press any key to continue . . .
```

Line: 1 Col: 19 Sel: 0 Lines: 24 Length: 525 Insert Done parsing in 0.156 seconds



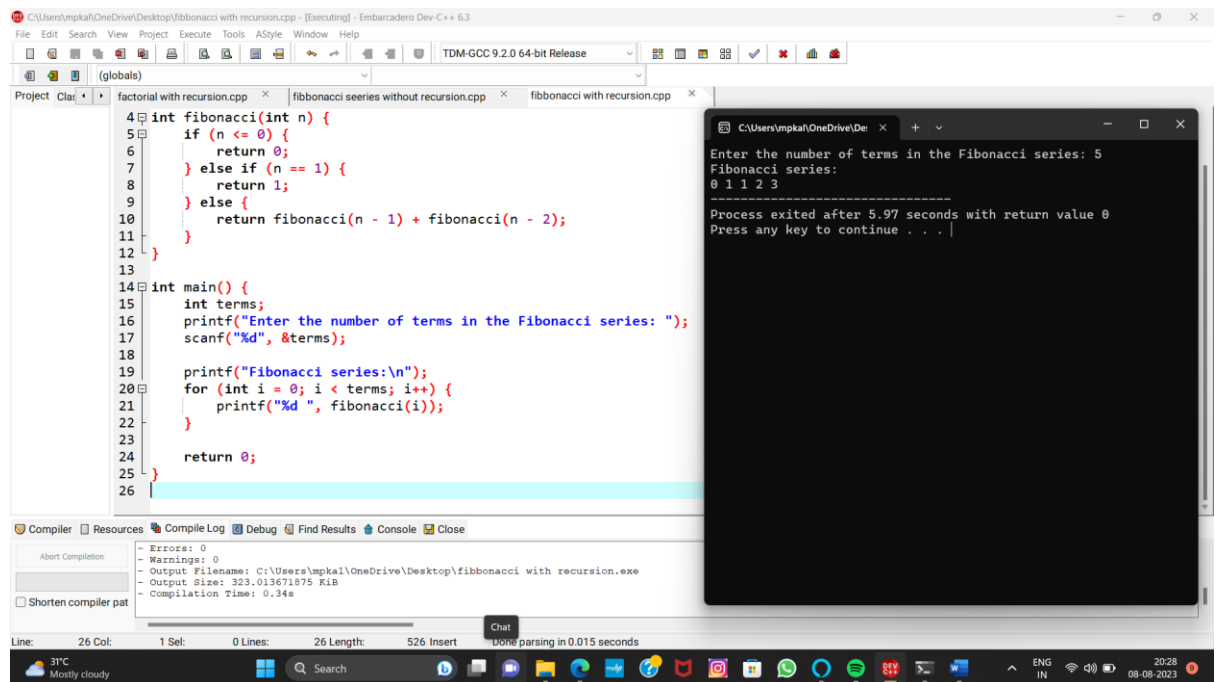
## 9.fibonacci without recursion

The screenshot shows a C++ IDE with a file named `fibonacci series without recursion.cpp`. The code implements a Fibonacci sequence using an iterative loop. The program prompts the user to enter the number of terms, which is 15. It then prints the sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377. The IDE's console window shows the execution output, including the prompt, the sequence, and a message indicating the process exited after 2.585 seconds with a return value of 0.

```
1 #include <stdio.h>
2
3 int main() {
4     int n, i;
5     long long int fib1 = 0, fib2 = 1, nextTerm;
6
7     printf("Enter the number of terms: ");
8     scanf("%d", &n);
9
10    printf("Fibonacci Series: %lld, %lld, ", fib1, fib2);
11
12    for (i = 3; i <= n; i++) {
13        nextTerm = fib1 + fib2;
14        printf("%lld, ", nextTerm);
15        fib1 = fib2;
16        fib2 = nextTerm;
17    }
18
19    printf("\n");
20
21    return 0;
22 }
23
```

Enter the number of terms: 15  
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377,  
-----  
Process exited after 2.585 seconds with return value 0  
Press any key to continue . . .

## 10.fibonacci with recursion



```
4 int fibonacci(int n) {
5     if (n <= 0) {
6         return 0;
7     } else if (n == 1) {
8         return 1;
9     } else {
10        return fibonacci(n - 1) + fibonacci(n - 2);
11    }
12 }
13
14 int main() {
15     int terms;
16     printf("Enter the number of terms in the Fibonacci series: ");
17     scanf("%d", &terms);
18
19     printf("Fibonacci series:\n");
20     for (int i = 0; i < terms; i++) {
21         printf("%d ", fibonacci(i));
22     }
23
24     return 0;
25 }
26
```

Compiler: Resources Compile Log Debug Find Results Console Close

Errors: 0  
Warnings: 0  
Output Filename: C:\Users\mpkal\OneDrive\Desktop\fibonacci with recursion.exe  
Output Size: 323,013671875 KiB  
Compilation Time: 0.34s

Line: 26 Col: 1 Sel: 0 Lines: 26 Length: 526 Insert Done parsing in 0.015 seconds

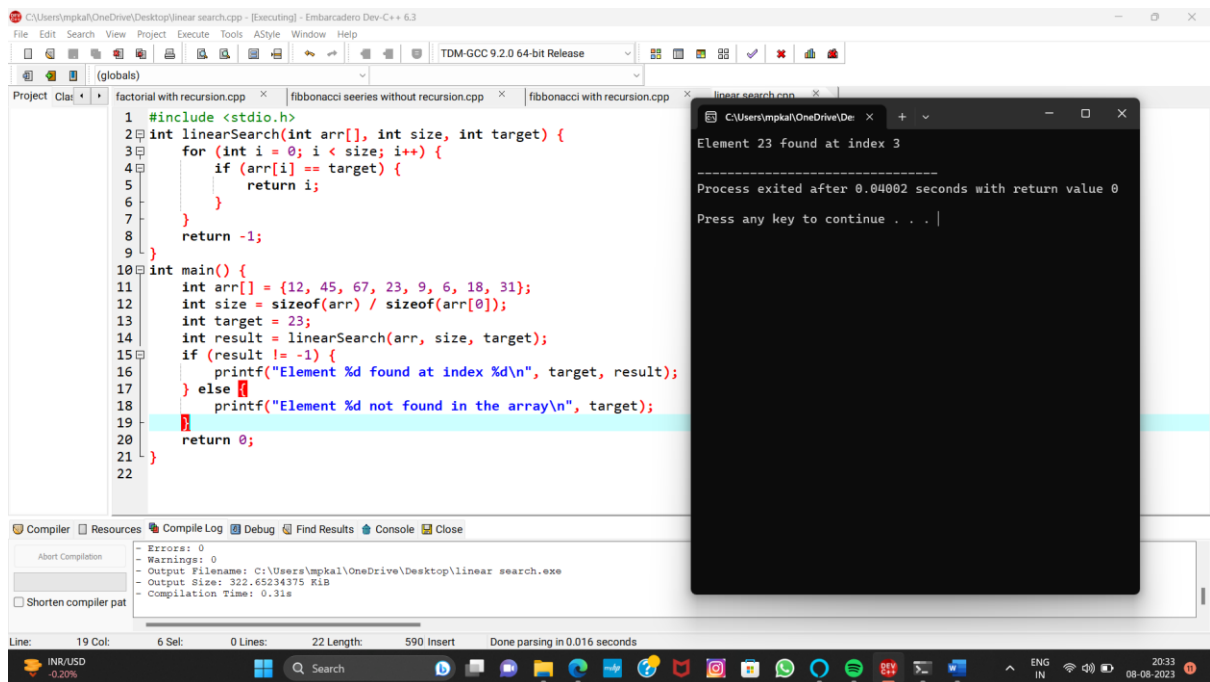
31°C Mostly cloudy

Chat

C:\Users\mpkal\OneDrive\De x + -

Enter the number of terms in the Fibonacci series: 5  
Fibonacci series:  
0 1 1 2 3  
-----  
Process exited after 5.97 seconds with return value 0  
Press any key to continue . . . |

# 11.linear search



The screenshot displays an IDE window with a C++ program for linear search. The code is as follows:

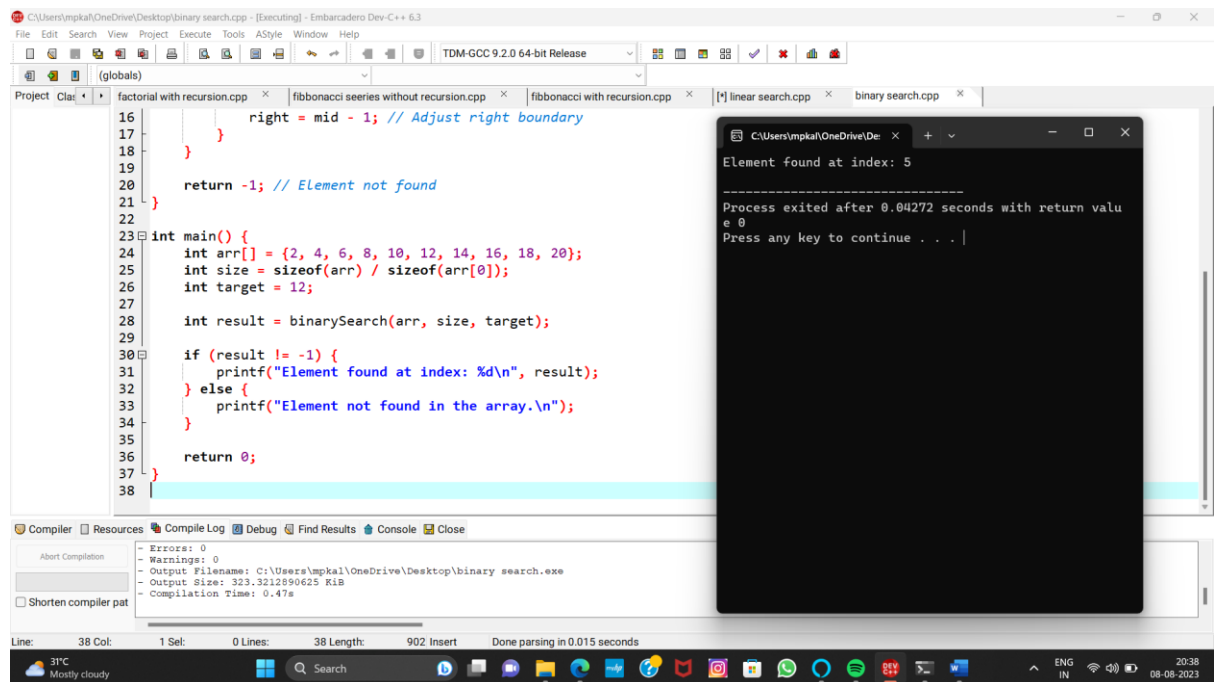
```
1 #include <stdio.h>
2 int linearSearch(int arr[], int size, int target) {
3     for (int i = 0; i < size; i++) {
4         if (arr[i] == target) {
5             return i;
6         }
7     }
8     return -1;
9 }
10 int main() {
11     int arr[] = {12, 45, 67, 23, 9, 6, 18, 31};
12     int size = sizeof(arr) / sizeof(arr[0]);
13     int target = 23;
14     int result = linearSearch(arr, size, target);
15     if (result != -1) {
16         printf("Element %d found at index %d\n", target, result);
17     } else {
18         printf("Element %d not found in the array\n", target);
19     }
20     return 0;
21 }
22
```

The IDE's console window shows the following output:

```
Element 23 found at index 3
-----
Process exited after 0.04002 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 status bar at the bottom showing line and column information (Line: 19, Col: 6) and compilation details.

## 12.binary search



The screenshot displays an IDE window titled "C:\Users\mpkal\OneDrive\Desktop\binary search.cpp - [Executing] - Embarcadero Dev-C++ 6.3". The main editor shows the source code for a binary search algorithm. The code includes a recursive function `binarySearch` and a `main` function that tests it with an array `{2, 4, 6, 8, 10, 12, 14, 16, 18, 20}` and a target value of 12. The output window on the right shows the program's execution, confirming that the element was found at index 5. The status bar at the bottom indicates the program was compiled successfully with no errors or warnings.

```
16         right = mid - 1; // Adjust right boundary
17     }
18 }
19
20     return -1; // Element not found
21 }
22
23 int main() {
24     int arr[] = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20};
25     int size = sizeof(arr) / sizeof(arr[0]);
26     int target = 12;
27
28     int result = binarySearch(arr, size, target);
29
30     if (result != -1) {
31         printf("Element found at index: %d\n", result);
32     } else {
33         printf("Element not found in the array.\n");
34     }
35
36     return 0;
37 }
38
```

Element found at index: 5

Process exited after 0.04272 seconds with return value 0  
Press any key to continue . . .

Compiler: Resources: Compile Log: Debug: Find Results: Console: Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\mpkal\OneDrive\Desktop\binary search.exe
- Output Size: 323.3212890625 KiB
- Compilation Time: 0.47s

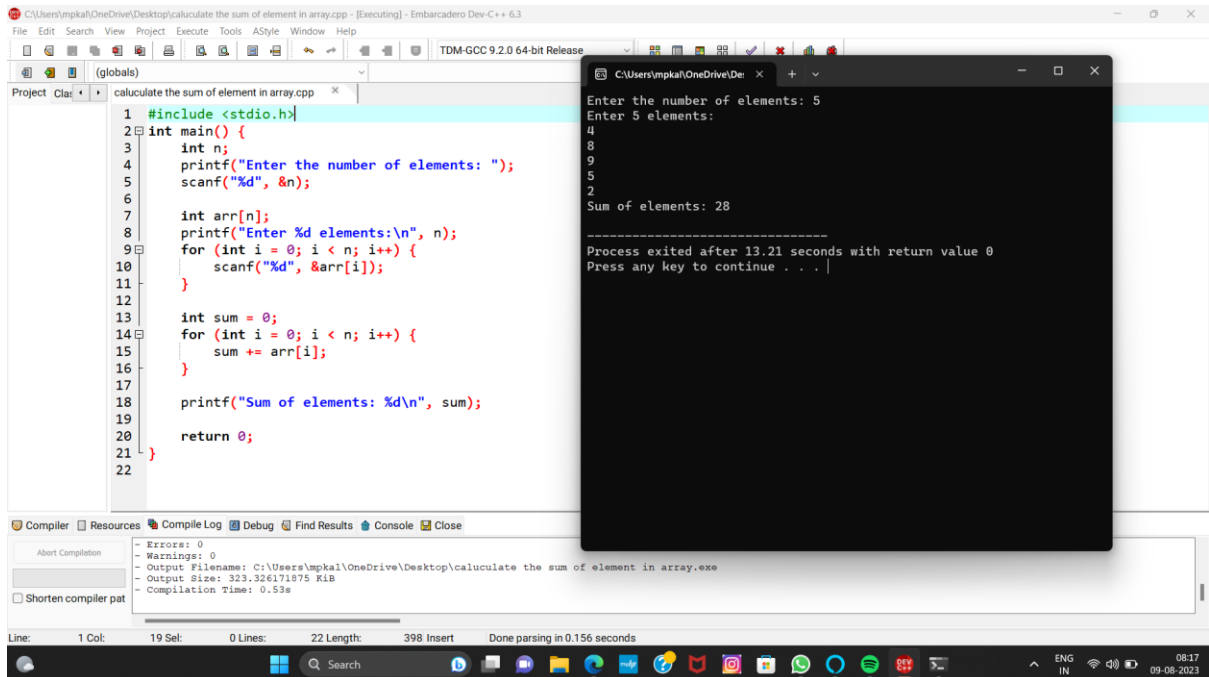
Line: 38 Col: 1 Sel: 0 Lines: 38 Length: 902 Insert Done parsing in 0.015 seconds

31°C Mostly cloudy

Search

ENG IN 20:38 08-08-2023

## 13.sum of elements in array

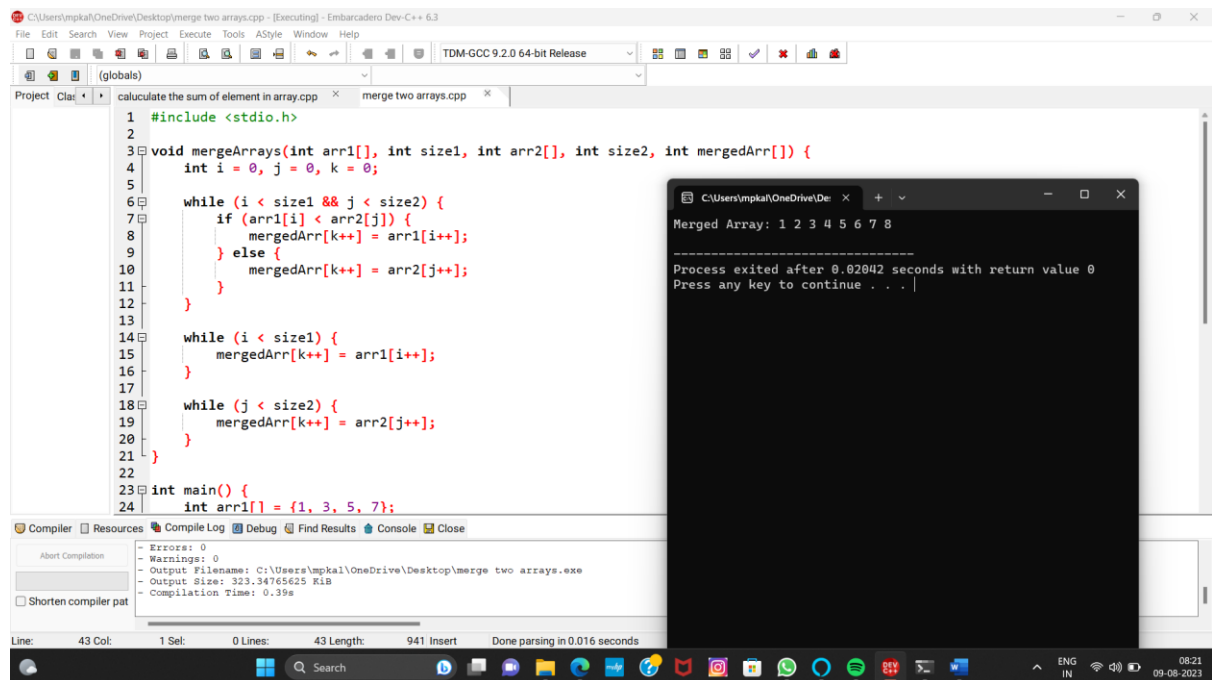


```
1 #include <stdio.h>
2 int main() {
3     int n;
4     printf("Enter the number of elements: ");
5     scanf("%d", &n);
6
7     int arr[n];
8     printf("Enter %d elements:\n", n);
9     for (int i = 0; i < n; i++) {
10         scanf("%d", &arr[i]);
11     }
12
13     int sum = 0;
14     for (int i = 0; i < n; i++) {
15         sum += arr[i];
16     }
17
18     printf("Sum of elements: %d\n", sum);
19
20     return 0;
21 }
22
```

Enter the number of elements: 5  
Enter 5 elements:  
4  
8  
9  
5  
2  
Sum of elements: 28  
-----  
Process exited after 13.21 seconds with return value 0  
Press any key to continue . . .

Compiler: TDM-GCC 9.2.0 64-bit Release  
Errors: 0  
Warnings: 0  
Output Filename: C:\Users\mpkal\OneDrive\Desktop\calculate the sum of element in array.exe  
Output Size: 323,326,171,975 KiB  
Compilation Time: 0.53s

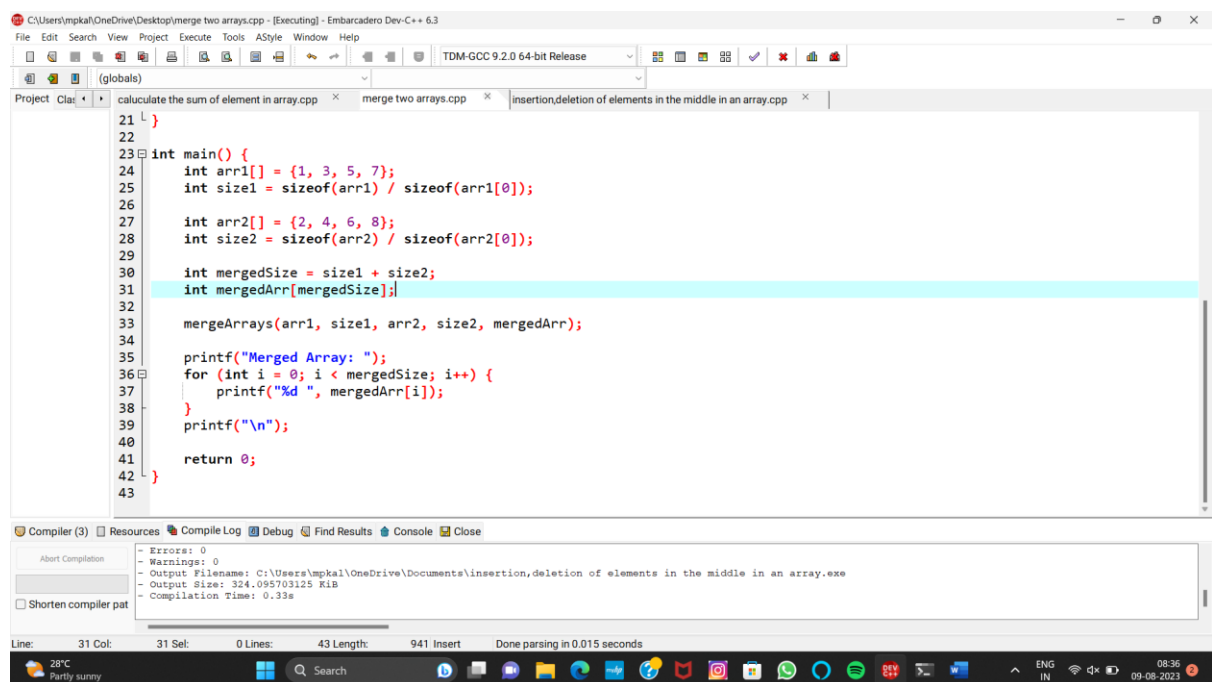
## 14.merge two arrays



The screenshot shows the Embarcadero Dev-C++ 6.3 IDE. The main editor window displays the code for 'merge two arrays.cpp'. The code defines a function 'mergeArrays' that takes two arrays and their sizes, and a 'mergedArr' array. It uses three while loops to merge the arrays: first, it merges the two arrays into 'mergedArr' using a comparison-based approach, then it copies the first array into 'mergedArr', and finally, it copies the second array into 'mergedArr'. The 'main' function initializes an array 'arr1' with values {1, 3, 5, 7} and calls 'mergeArrays'.

```
1 #include <stdio.h>
2
3 void mergeArrays(int arr1[], int size1, int arr2[], int size2, int mergedArr[]) {
4     int i = 0, j = 0, k = 0;
5
6     while (i < size1 && j < size2) {
7         if (arr1[i] < arr2[j]) {
8             mergedArr[k++] = arr1[i++];
9         } else {
10            mergedArr[k++] = arr2[j++];
11        }
12    }
13
14    while (i < size1) {
15        mergedArr[k++] = arr1[i++];
16    }
17
18    while (j < size2) {
19        mergedArr[k++] = arr2[j++];
20    }
21 }
22
23 int main() {
24     int arr1[] = {1, 3, 5, 7};
```

The console window shows the output of the program: 'Merged Array: 1 2 3 4 5 6 7 8'. Below the output, it states 'Process exited after 0.02842 seconds with return value 0' and 'Press any key to continue . . .'. The status bar at the bottom indicates 'Line: 43 Col: 1 Sel: 0 Lines: 43 Length: 941 Insert Done parsing in 0.016 seconds'.

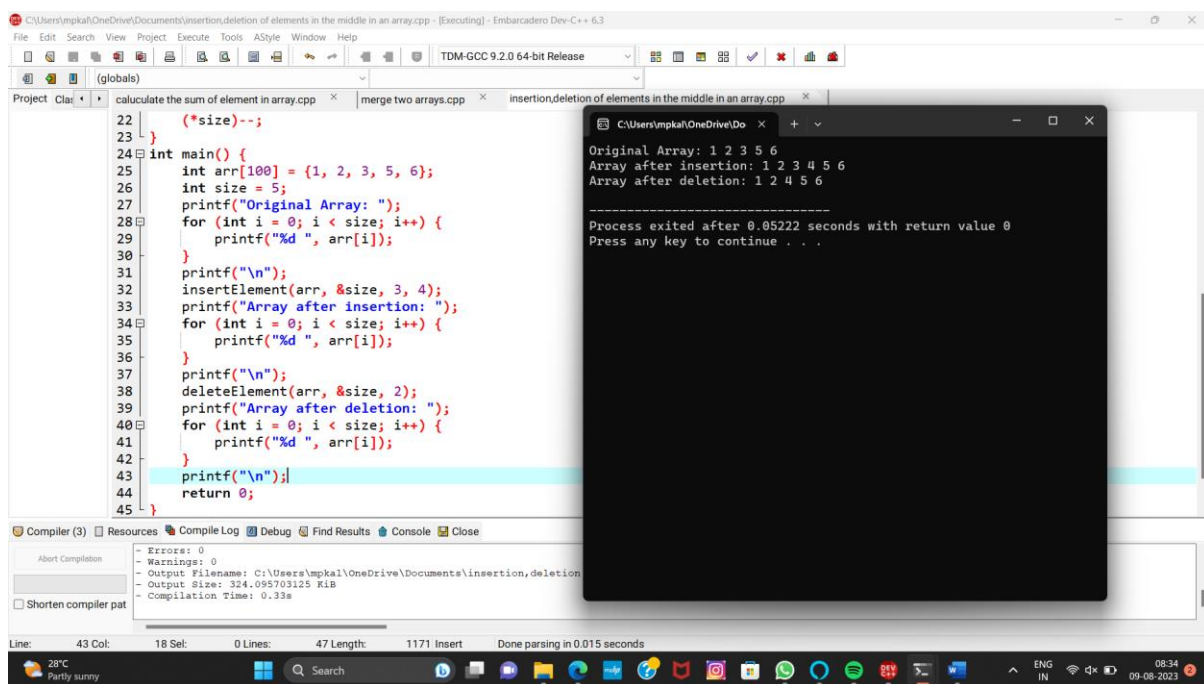


The screenshot shows the Embarcadero Dev-C++ 6.3 IDE. The main editor window displays the code for 'insertion, deletion of elements in the middle in an array.cpp'. The code defines a function 'insertionDeletion' that takes an array, its size, and a new element to be inserted. It calculates the 'mergedSize' as the sum of the original size and the size of the new element. The 'main' function initializes two arrays, 'arr1' and 'arr2', and calls 'insertionDeletion' to insert the elements of 'arr2' into 'arr1'.

```
21 }
22
23 int main() {
24     int arr1[] = {1, 3, 5, 7};
25     int size1 = sizeof(arr1) / sizeof(arr1[0]);
26
27     int arr2[] = {2, 4, 6, 8};
28     int size2 = sizeof(arr2) / sizeof(arr2[0]);
29
30     int mergedSize = size1 + size2;
31     int mergedArr[mergedSize];
32
33     mergeArrays(arr1, size1, arr2, size2, mergedArr);
34
35     printf("Merged Array: ");
36     for (int i = 0; i < mergedSize; i++) {
37         printf("%d ", mergedArr[i]);
38     }
39     printf("\n");
40
41     return 0;
42 }
43 }
```

The console window shows the output of the program: 'Merged Array: 1 2 3 4 5 6 7 8'. Below the output, it states 'Process exited after 0.02842 seconds with return value 0' and 'Press any key to continue . . .'. The status bar at the bottom indicates 'Line: 31 Col: 31 Sel: 0 Lines: 43 Length: 941 Insert Done parsing in 0.015 seconds'.

## 15.insertion,deletion in the middle of array



The screenshot shows an IDE with a C++ project titled "insertion,deletion of elements in the middle in an array.cpp". The code implements a program to insert and delete elements from an array. The console window displays the output of the program.

```
22 } (*size)--;
23 }
24 int main() {
25     int arr[100] = {1, 2, 3, 5, 6};
26     int size = 5;
27     printf("Original Array: ");
28     for (int i = 0; i < size; i++) {
29         printf("%d ", arr[i]);
30     }
31     printf("\n");
32     insertElement(arr, &size, 3, 4);
33     printf("Array after insertion: ");
34     for (int i = 0; i < size; i++) {
35         printf("%d ", arr[i]);
36     }
37     printf("\n");
38     deleteElement(arr, &size, 2);
39     printf("Array after deletion: ");
40     for (int i = 0; i < size; i++) {
41         printf("%d ", arr[i]);
42     }
43     printf("\n");
44     return 0;
45 }
```

Compiler (3) | Resources | Compile Log | Debug | Find Results | Console | Close

Errors: 0  
Warnings: 0  
Output Filename: C:\Users\mpkal\OneDrive\Documents\insertion,deletion of elements in the middle in an array.cpp  
Output Size: 324.095703125 KiB  
Compilation Time: 0.33s

Line: 43 Col: 18 Sel: 0 Lines: 47 Length: 1171 Insert Done parsing in 0.015 seconds

23°C Partly sunny

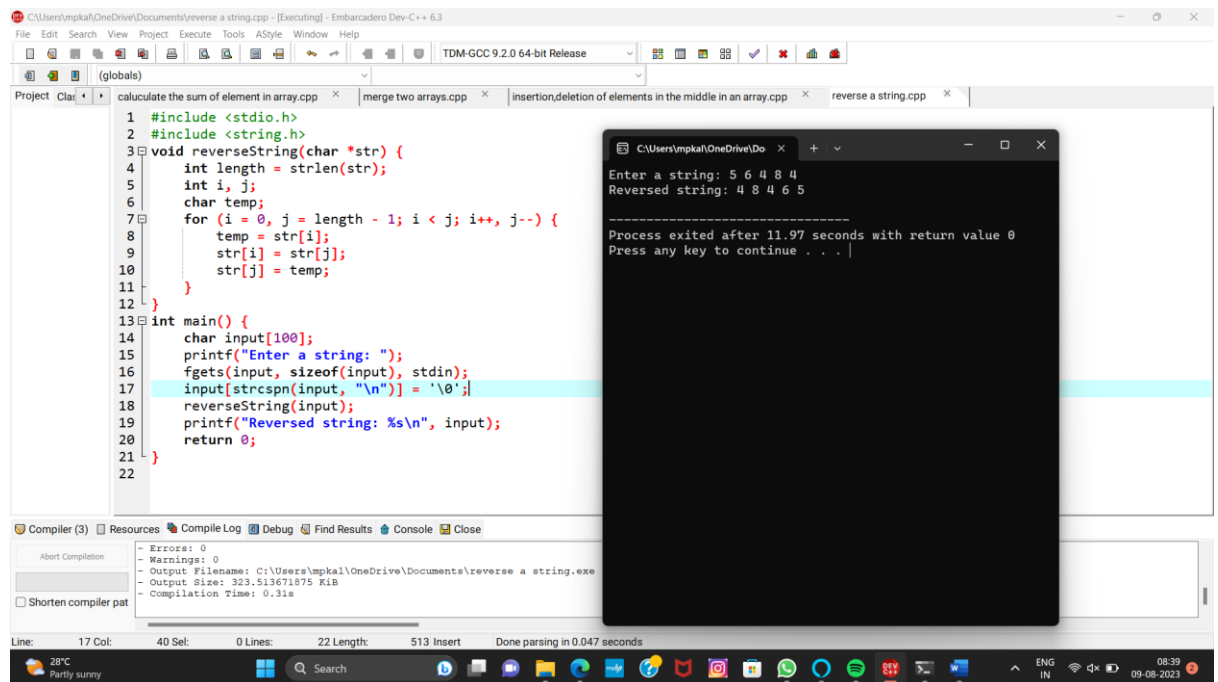
ENG IN 08:34 09-08-2023

Console Output:

```
C:\Users\mpkal\OneDrive\Documents\insertion,deletion of elements in the middle in an array.cpp
Original Array: 1 2 3 5 6
Array after insertion: 1 2 3 4 5 6
Array after deletion: 1 2 4 5 6

Process exited after 0.05222 seconds with return value 0
Press any key to continue . . .
```

## 16.reverse a string



The screenshot displays a C++ IDE with a project named "reverse a string.cpp". The code implements a function to reverse a string using a two-pointer technique. The main function prompts the user to enter a string, reads it, and prints the reversed string. The output window shows the execution results for the input "5 6 4 8 4", resulting in the reversed string "4 8 4 6 5".

```
1 #include <stdio.h>
2 #include <string.h>
3 void reverseString(char *str) {
4     int length = strlen(str);
5     int i, j;
6     char temp;
7     for (i = 0, j = length - 1; i < j; i++, j--) {
8         temp = str[i];
9         str[i] = str[j];
10        str[j] = temp;
11    }
12 }
13 int main() {
14     char input[100];
15     printf("Enter a string: ");
16     fgets(input, sizeof(input), stdin);
17     input[strcspn(input, "\n")] = '\0';
18     reverseString(input);
19     printf("Reversed string: %s\n", input);
20     return 0;
21 }
22
```

Compiler (3) | Resources | Compile Log | Debug | Find Results | Console | Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\mpkal\OneDrive\Documents\reverse a string.exe
- Output Size: 323,513,671,875 KiB
- Compilation Time: 0.31s

Shorten compiler pat

Line: 17 Col: 40 Sel: 0 Lines: 22 Length: 513 Insert Done parsing in 0.047 seconds

28°C Partly sunny

Search

ENG IN 08:39 09-08-2023

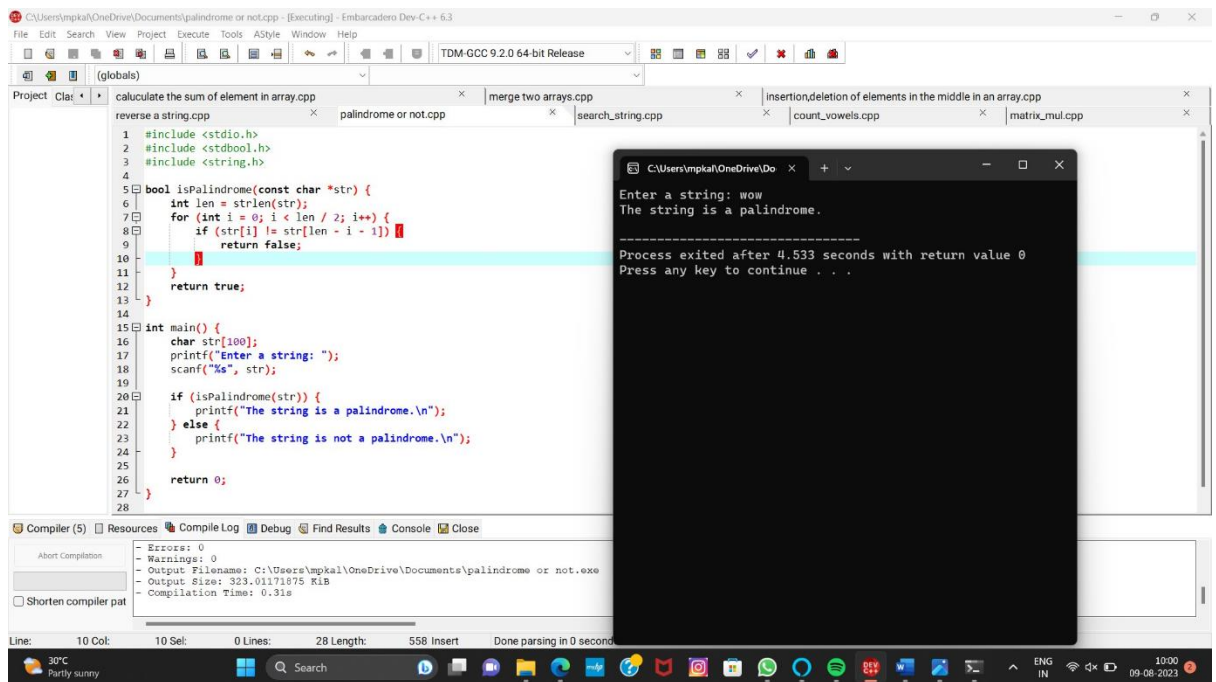
Output Window:

```
C:\Users\mpkal\OneDrive\Do
Enter a string: 5 6 4 8 4
Reversed string: 4 8 4 6 5

Process exited after 11.97 seconds with return value 0
Press any key to continue . . .
```



## 17.palindrome or not



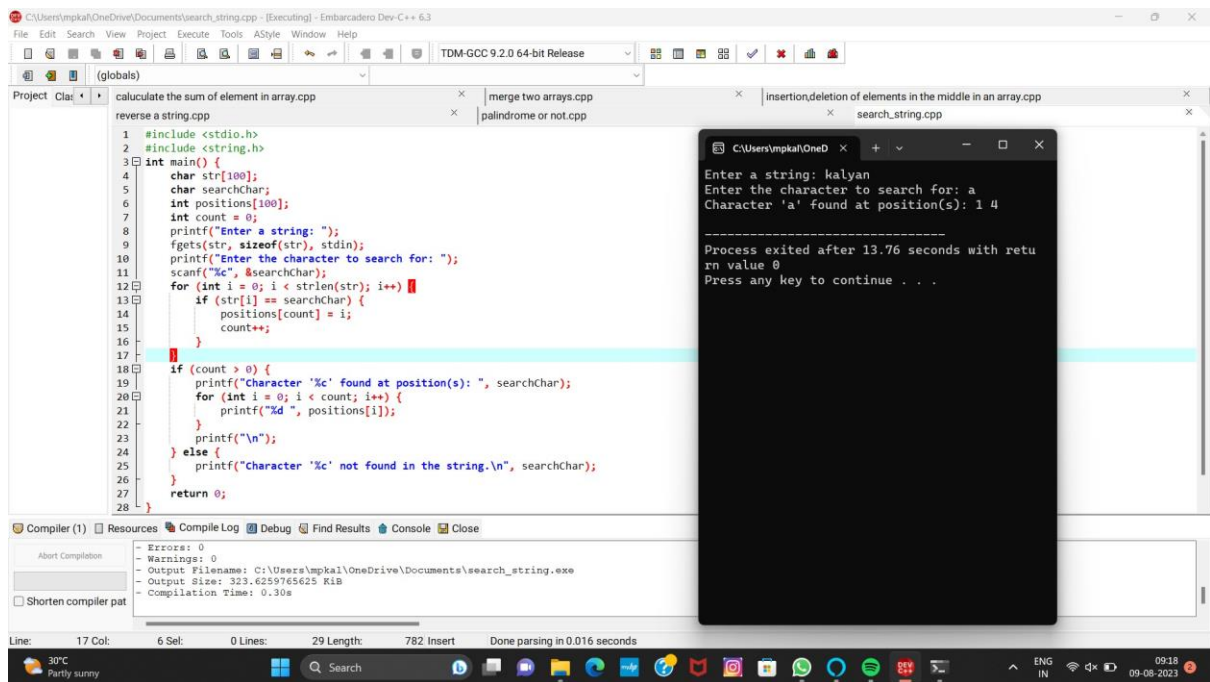
The screenshot shows a C++ IDE with a project named "palindrome or not.cpp". The code defines a function `isPalindrome` that checks if a string is a palindrome by comparing characters from both ends. The `main` function prompts the user to enter a string and prints the result.

```
1 #include <stdio.h>
2 #include <stdbool.h>
3 #include <string.h>
4
5 bool isPalindrome(const char *str) {
6     int len = strlen(str);
7     for (int i = 0; i < len / 2; i++) {
8         if (str[i] != str[len - i - 1])
9             return false;
10    }
11    return true;
12 }
13
14 int main() {
15     char str[100];
16     printf("Enter a string: ");
17     scanf("%s", str);
18
19     if (isPalindrome(str)) {
20         printf("The string is a palindrome.\n");
21     } else {
22         printf("The string is not a palindrome.\n");
23     }
24
25     return 0;
26 }
```

The console output shows the program running with the input "wow". The output is "The string is a palindrome." and the process exits after 4.533 seconds with return value 0.

```
Enter a string: wow
The string is a palindrome.
-----
Process exited after 4.533 seconds with return value 0
Press any key to continue . . .
```

## 18. search a particular character in a string



The screenshot displays an IDE window titled "C:\Users\mpkal\OneDrive\Documents\search\_string.cpp - [Executing] - Embarcadero Dev-C++ 6.3". The code in the editor is as follows:

```
1 #include <stdio.h>
2 #include <string.h>
3 int main() {
4     char str[100];
5     char searchChar;
6     int positions[100];
7     int count = 0;
8     printf("Enter a string: ");
9     fgets(str, sizeof(str), stdin);
10    printf("Enter the character to search for: ");
11    scanf("%c", &searchChar);
12    for (int i = 0; i < strlen(str); i++) {
13        if (str[i] == searchChar) {
14            positions[count] = i;
15            count++;
16        }
17    }
18    if (count > 0) {
19        printf("Character '%c' found at position(s): ", searchChar);
20        for (int i = 0; i < count; i++) {
21            printf("%d ", positions[i]);
22        }
23        printf("\n");
24    } else {
25        printf("Character '%c' not found in the string.\n", searchChar);
26    }
27    return 0;
28 }
```

The output window shows the following execution results:

```
Enter a string: kalyan
Enter the character to search for: a
Character 'a' found at position(s): 1 4

-----
Process exited after 13.76 seconds with return value 0
Press any key to continue . . .
```

The IDE's status bar at the bottom indicates "Line: 17 Col: 6 Sel: 0 Lines: 29 Length: 782 Insert Done parsing in 0.016 seconds". The taskbar at the very bottom shows the system clock as 09:18 on 09-08-2023.

# 19.count vowels

```
1 #include <stdio.h>
2 int main()
3 {
4     int c = 0, count = 0;
5     char s[1000];
6     printf("Input a string\n");
7     gets(s);
8
9
10    while (s[c] != '\0') {
11        if (s[c] == 'a' || s[c] == 'e' || s[c] == 'i' || s[c] == 'o' || s[c] == 'u')
12            count++;
13        c++;
14    }
15    printf("Number of vowels in the string: %d", count);
16    return 0;
17 }
```

Compiler (5) Resources Compile Log Debug Find Results Console Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\mpka1\OneDrive\Documents\count\_vowels.exe
- Output Size: 323,452,146,4375 KiB
- Compilation Time: 0.31s

Line: 17 Col: 10 Sel: 0 Lines: 18 Length: 405 Insert Done parsing in 0.015 seconds

30°C Partly sunny

Search

ENG IN 09:23 09-08-2023

```
C:\Users\mpka1\OneD x + - _ □ x
Input a string
poornakalyan
Number of vowels in the string: 5
-----
Process exited after 8.064 seconds with return value 0
Press any key to continue . . .
```

## 20.matrix multiplication

The screenshot shows a C++ IDE with the following components:

- Project Explorer:** Lists files including `calculate the sum of element in array.cpp`, `reverse a string.cpp`, `palindrome or not.cpp`, `merge two arrays.cpp`, `insertion, deletion of elements in the middle in an array.cpp`, and `_mul.cpp`.
- Editor:** Displays the code for `_mul.cpp`, which includes functions for matrix multiplication and printing.
- Compiler Output:** Shows the compilation process with the following details:
  - Errors: 0
  - Warnings: 0
  - Output Filename: `C:\Users\mpkal\OneDrive\Documents\matrix_mul.exe`
  - Output Size: `324.224609375 KiB`
  - Compilation Time: `0.31s`
- Console:** Displays the program's execution output:

```
Enter the number of rows and columns of matrix A: 2
2
Enter the number of rows and columns of matrix B: 2
2
Enter elements of matrix A:
1
2
3
4
Enter elements of matrix B:
7
8
9
4
4
Matrix A:
1 2
3 4
Matrix B:
7 8
9 4
4
Resultant matrix after multiplication:
25 16
57 40

Process exited after 24.37 seconds with return value 0
Press any key to continue . . .
```