

1. Write a C program to arrange numbers using Selection Sort.

Aim: To write a C program to sort a given list of numbers using Selection Sort .

Algorithm:

1. Take numbers in an array
2. Find the smallest number and put it first
3. Repeat for the remaining numbers
4. Print the sorted array

Input: 29,10,14,37,13

Output: 10 13 14 29 37

The screenshot displays an IDE window titled 'D:\Untitled2.cpp - [Executing] - Embarcadero Dev-C++ 6.3'. The code editor shows a C program that calculates the factorial of a number. The code is as follows:

```
1 #include <stdio.h>
2 int factorial(int n) {
3     if(n <= 1)
4         return 1;
5     return n * factorial(n - 1);
6 }
7 int main() {
8     int num;
9     printf("Enter a number: ");
10    scanf("%d", &num);
11    printf("Factorial: %d\n", factorial(num));
12    return 0;
13 }
14
```

The console window, titled 'D:\Untitled2.exe', shows the program's execution. It prompts the user to 'Enter a number: 5' and outputs 'Factorial: 120'. Below the output, it states 'Process exited after 9.407 seconds with return value 0' and 'Press any key to continue . . .'. The status bar at the bottom of the IDE indicates 'Line: 13 Col: 2 Sel: 0 Lines: 14 Length: 267 Insert Done parsing in 0.031 seconds'. The system tray at the bottom shows the date and time as '14-10-2025 15:23'.

2. Duplicate in a instruction.

Aim:

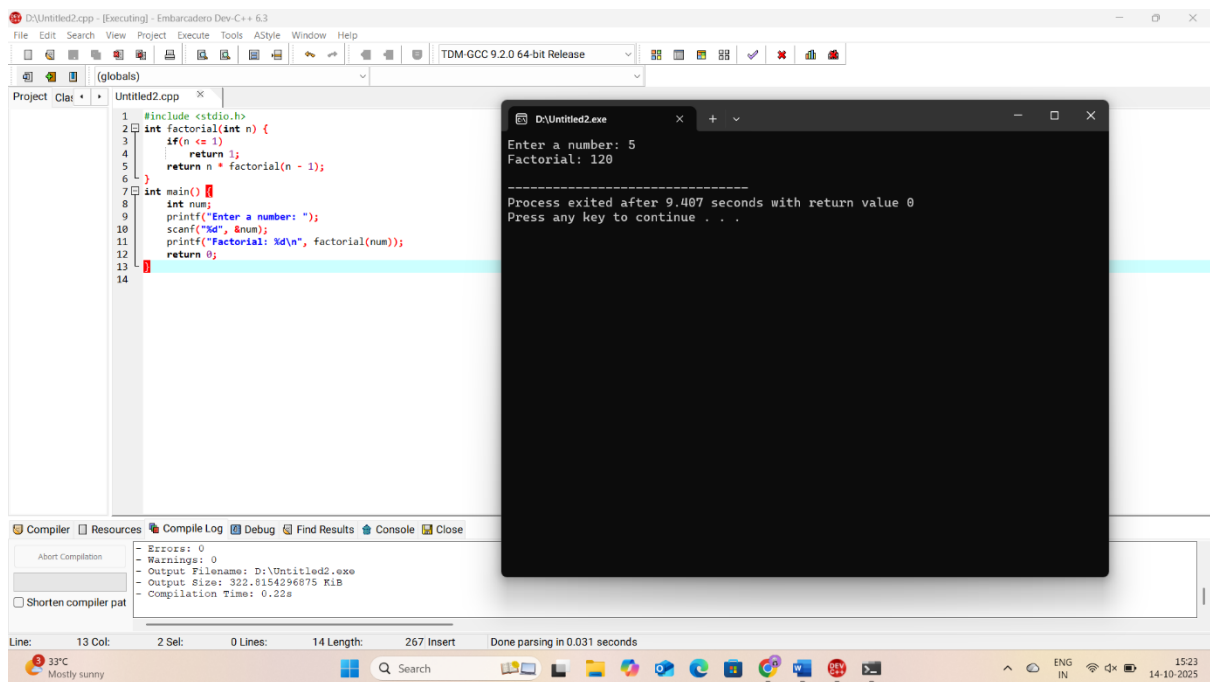
To write a C program to find duplicate elements in an array.

Algorithm:

1. Start
2. Read n numbers into array
3. Compare each element with others
4. If any two are equal, print as duplicate
5. Stop

Input: number=5

Output: 120



The screenshot displays an IDE window titled "D:\Untitled2.cpp - [Executing] - Embarcadero Dev-C++ 6.3". The code editor shows a C program for calculating the factorial of a number. The code is as follows:

```
1 #include <stdio.h>
2 int factorial(int n) {
3     if(n <= 1)
4         return 1;
5     return n * factorial(n - 1);
6 }
7 int main() {
8     int num;
9     printf("Enter a number: ");
10    scanf("%d", &num);
11    printf("Factorial: %d\n", factorial(num));
12    return 0;
13 }
14
```

The output window, titled "D:\Untitled2.exe", shows the execution results:

```
Enter a number: 5
Factorial: 120

-----
Process exited after 9.487 seconds with return value 0
Press any key to continue . . .
```

The bottom status bar indicates "Line: 13 Col: 2 Sel: 0 Lines: 14 Length: 267 Insert Done parsing in 0.031 seconds". The system tray at the bottom shows a temperature of 33°C, weather "Mostly sunny", and the date "14-10-2025".

3. Bigger Number in a Series.

Aim:

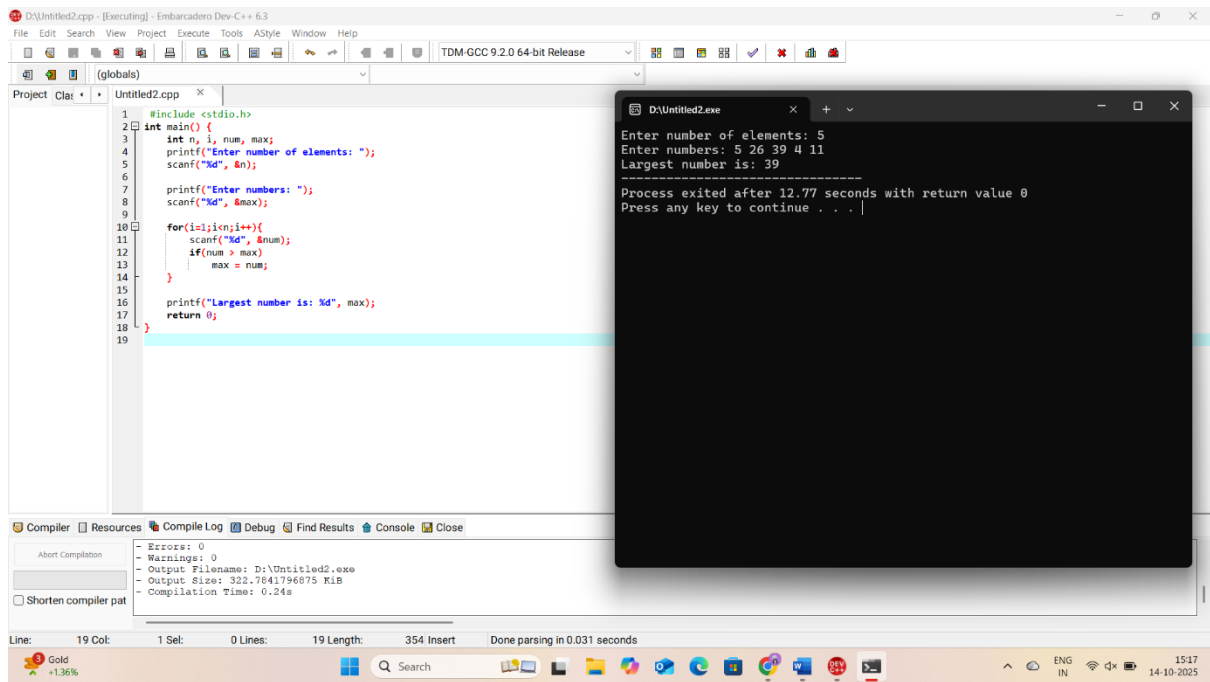
To write a C program to find the largest number from given numbers.

Algorithm:

1. Start
2. Read n numbers
3. Assume first number as max
4. Compare each number with max
5. If bigger, update max
6. Print max
7. Stop

Input: 5 26 39 4 11

Output: 39



4. Recursion – Factorial of a Given Number.

Aim:

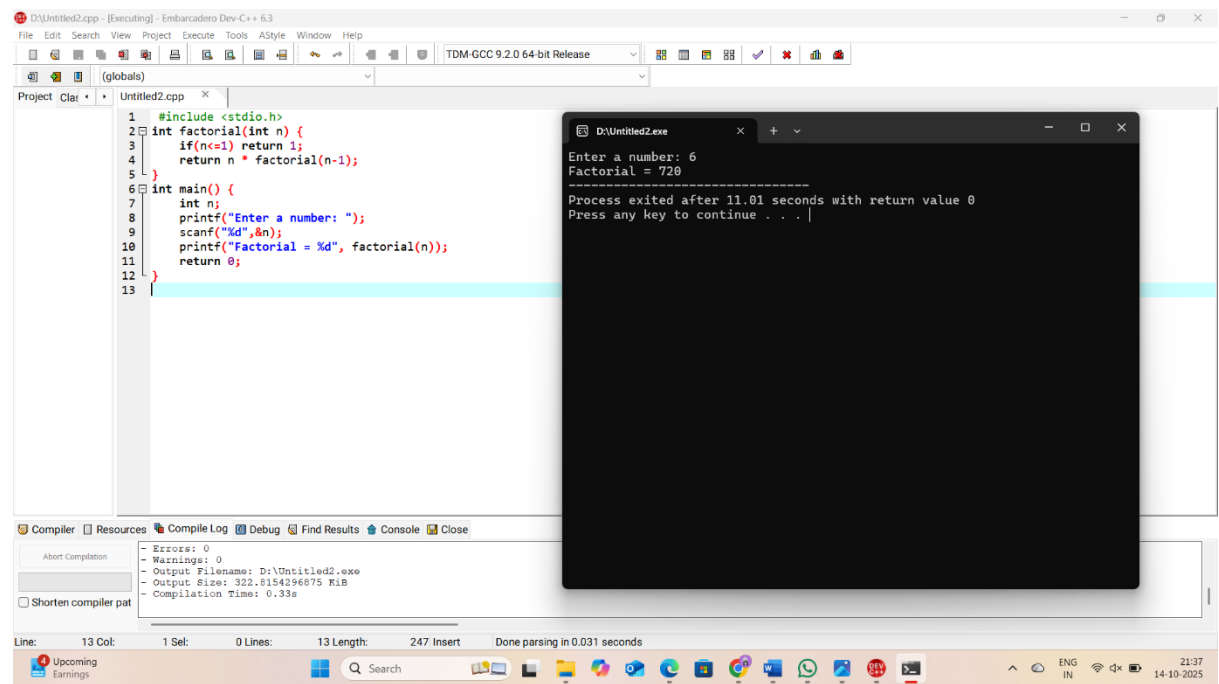
To write a C program to find the factorial of a number using recursion.

Algorithm:

1. Start
2. Read a number n
3. If $n==0$ or $n==1 \rightarrow$ return 1
4. Else return $n * \text{factorial}(n-1)$
5. Print result
6. Stop

Input: 6

Output: 720



5. Fibonacci Series.

Aim:

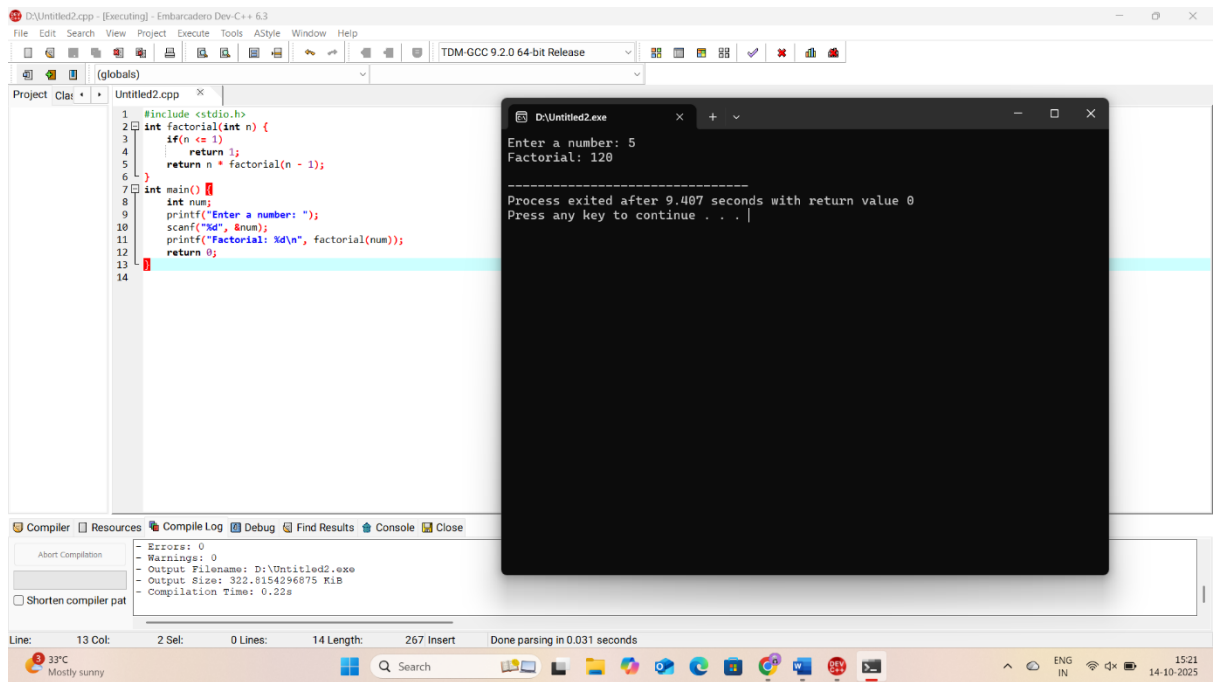
To write a C program to generate the Fibonacci series.

Algorithm:

1. Start
2. Read n terms
3. Initialize t1=0, t2=1
4. Print t1 and t2
5. Repeat for remaining terms: next = t1+t2, print, update t1=t2, t2=next
6. Stop

Input: 5

Output: 120



6. Two Order Homogeneous Recursion.

Aim:

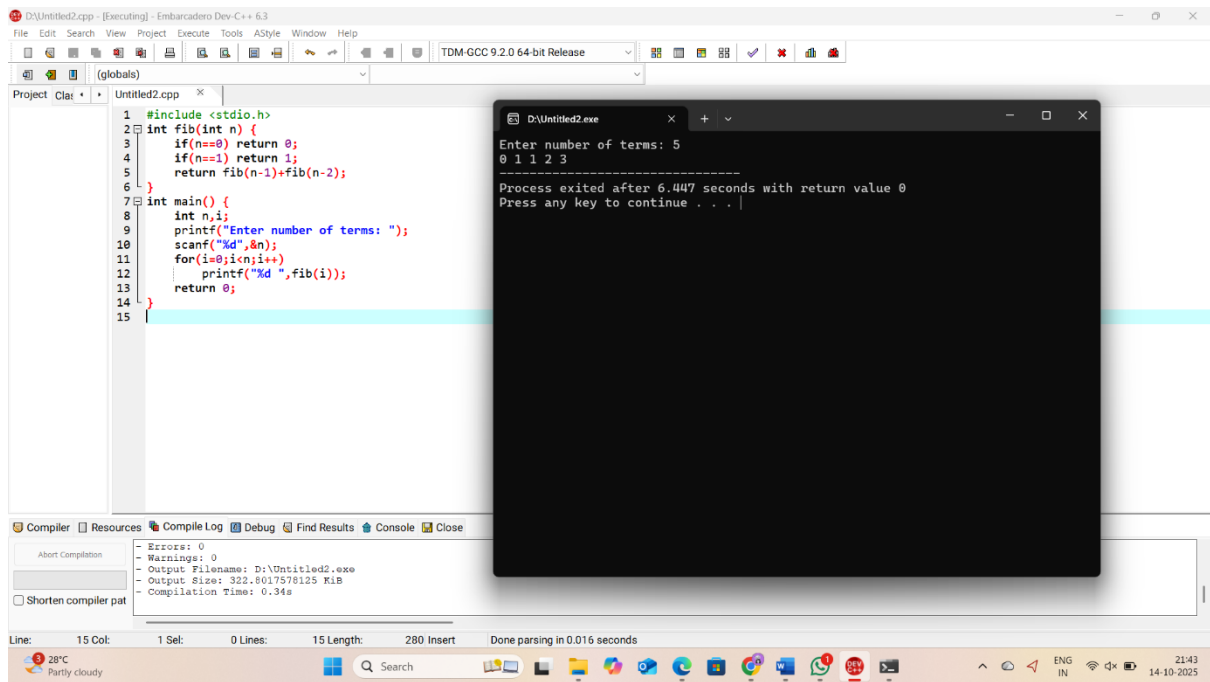
To write a C program using recursion for a second-order homogeneous recurrence relation.

Algorithm:

1. Start
2. Define recursive relation: $F(n)=F(n-1)+F(n-2)$
3. Base cases: $F(0)=0, F(1)=1$
4. Print terms using recursion
5. Stop

Input: terms=5

Output: 0 1 1 2 3



7. Leap Year

Aim:

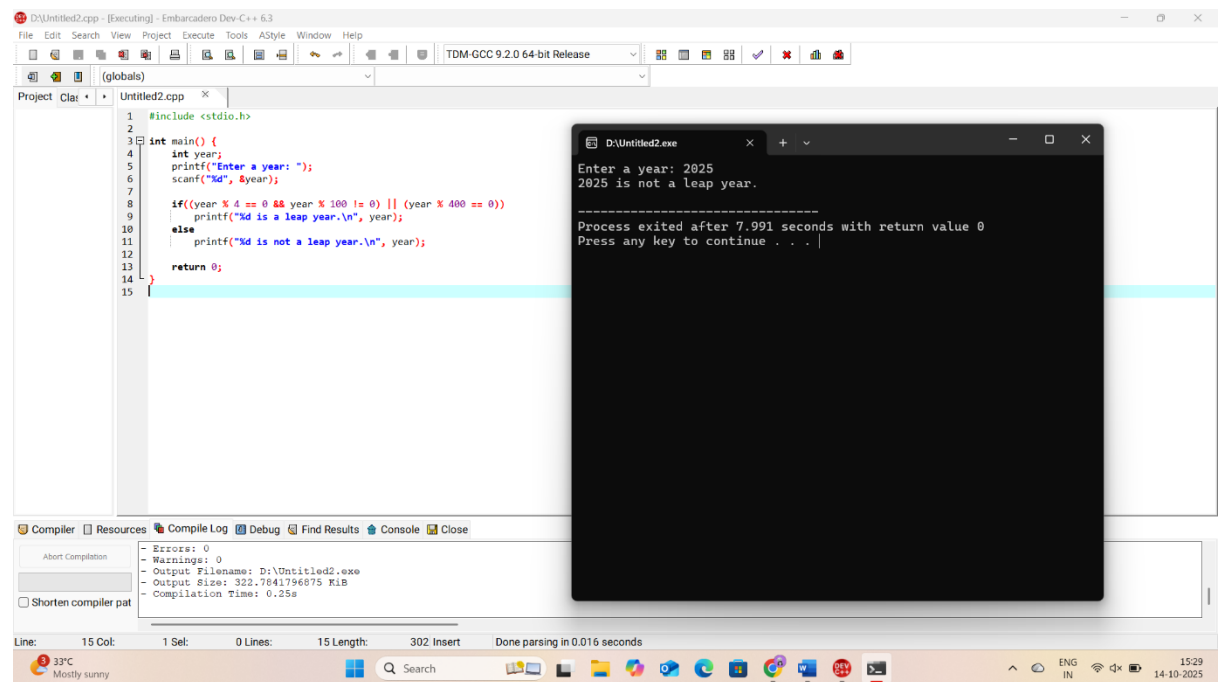
To write a C program to check whether a year is a leap year.

Algorithm:

1. Start
2. Read year
3. If divisible by 400 → leap year
4. Else if divisible by 4 but not by 100 → leap year
5. Else not a leap year
6. Stop

Input: year=2025

Output: 2025 is not a leap year



8. Swapping of Numbers.

Aim:

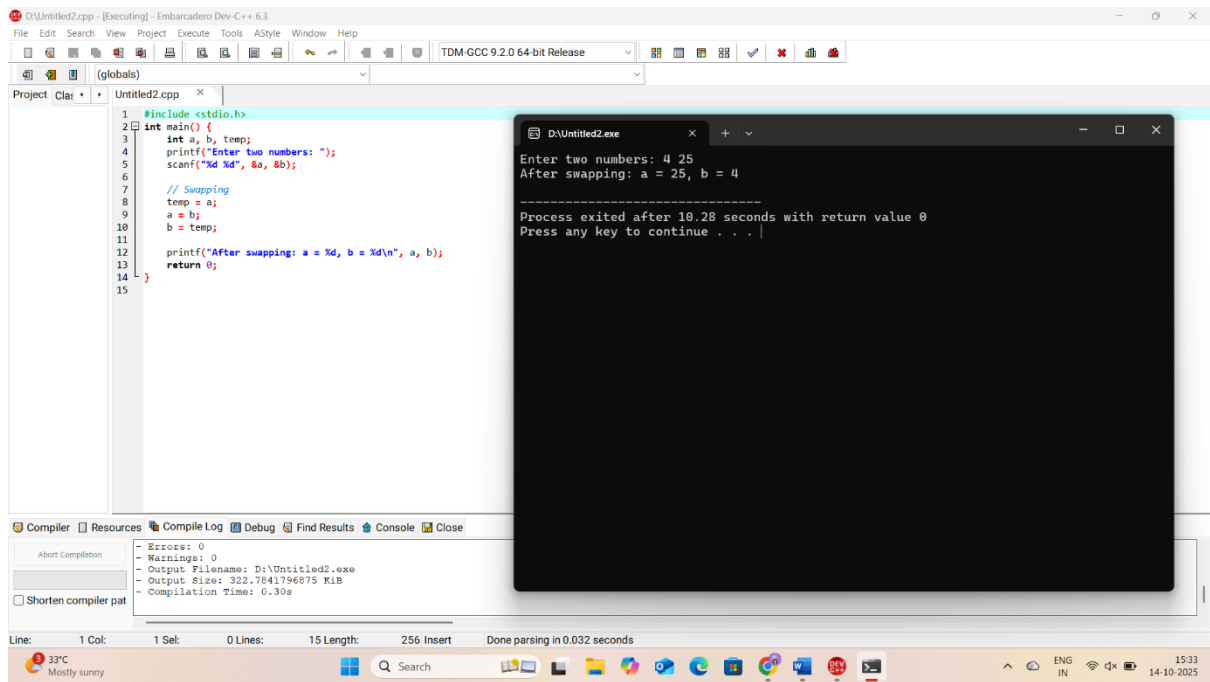
To write a C program to swap two numbers.

Algorithm:

1. Start
2. Read two numbers a and b
3. Swap using temp variable (or without)
4. Print swapped values
5. Stop

Input: a=4 b=25

Output: a=25 b=4



9. Identifying Palindrome

Aim:

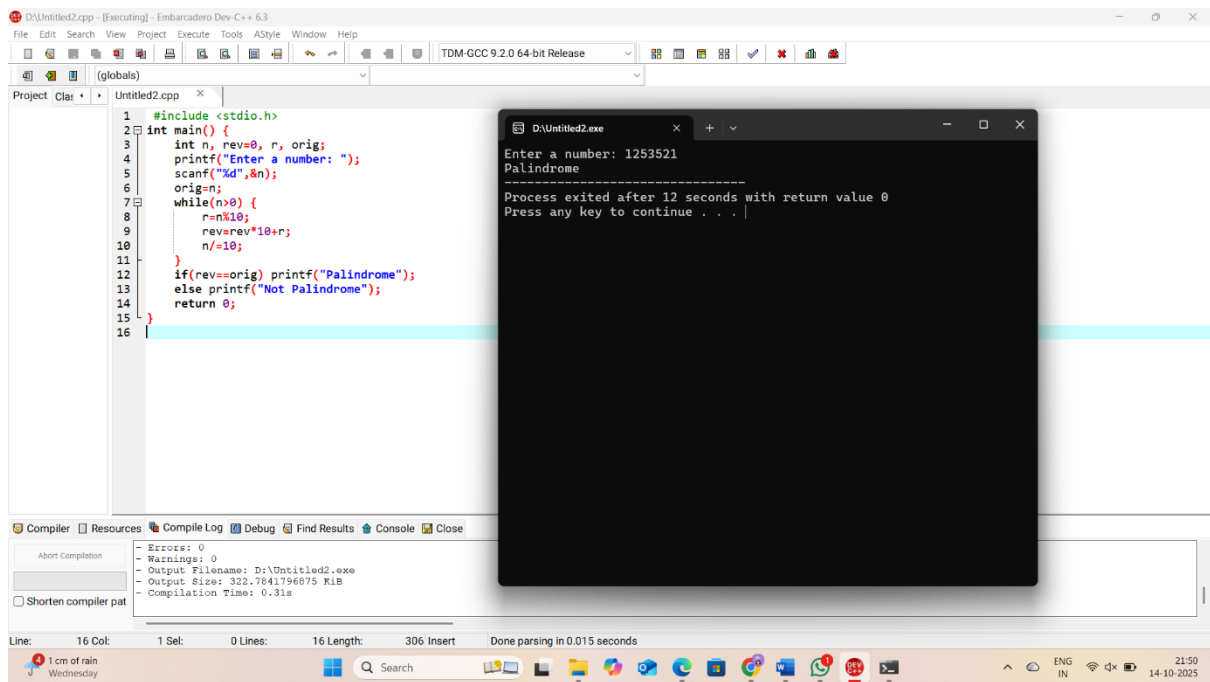
To write a C program to check whether a number is a palindrome.

Algorithm:

1. Start
2. Read a number n
3. Reverse digits of n
4. If reverse = original → palindrome
5. Else not palindrome
6. Stop

Input: 1253521

Output: palindrome



10. Prime Number

Aim:

To write a C program to check whether a number is prime.

Algorithm:

1. Start
2. Read n
3. If $n \leq 1 \rightarrow$ not prime
4. Check divisibility from 2 to $n/2$
5. If divisible \rightarrow not prime
6. Else \rightarrow prime
7. Stop

Input: 53

Output: prime number

