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

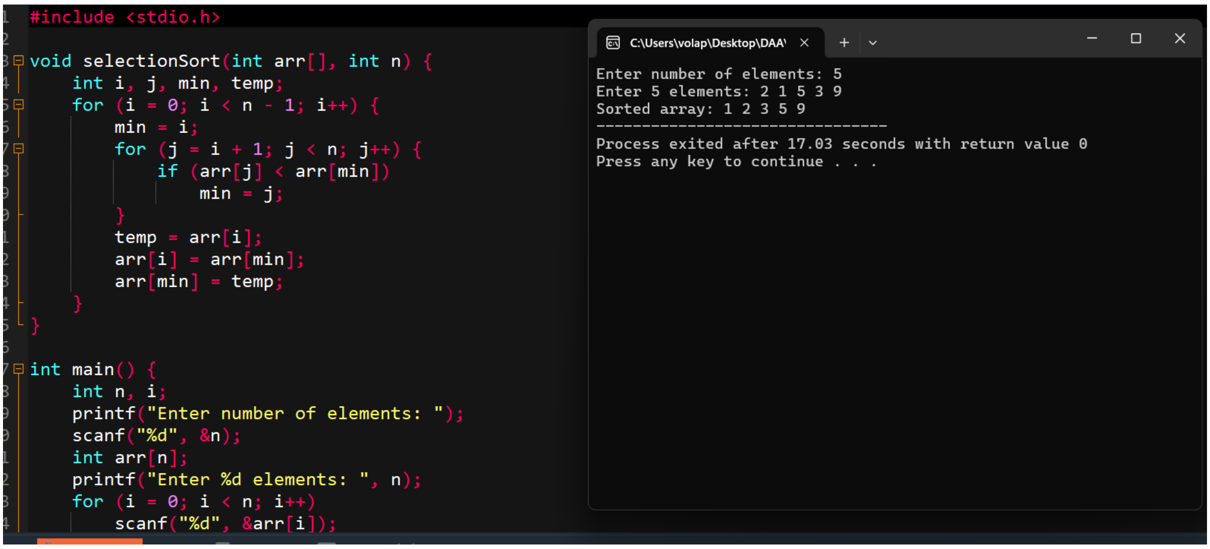
Aim: To write a C program to sort a given list of numbers using Selection Sort .

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: 2,1,5,3,9

Output: 1 2 3 5 9



```c
#include <stdio.h>

void selectionSort(int arr[], int n) {
    int i, j, min, temp;
    for (i = 0; i < n - 1; i++) {
        min = i;
        for (j = i + 1; j < n; j++) {
            if (arr[j] < arr[min])
                min = j;
        }
        temp = arr[i];
        arr[i] = arr[min];
        arr[min] = temp;
    }
}

int main() {
    int n, i;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
```

```
Enter number of elements: 5
Enter 5 elements: 2 1 5 3 9
Sorted array: 1 2 3 5 9
--------------------------------
Process exited after 17.03 seconds with return value 0
Press any key to continue . . .
```

2. Duplicate in a instruction.

Aim:

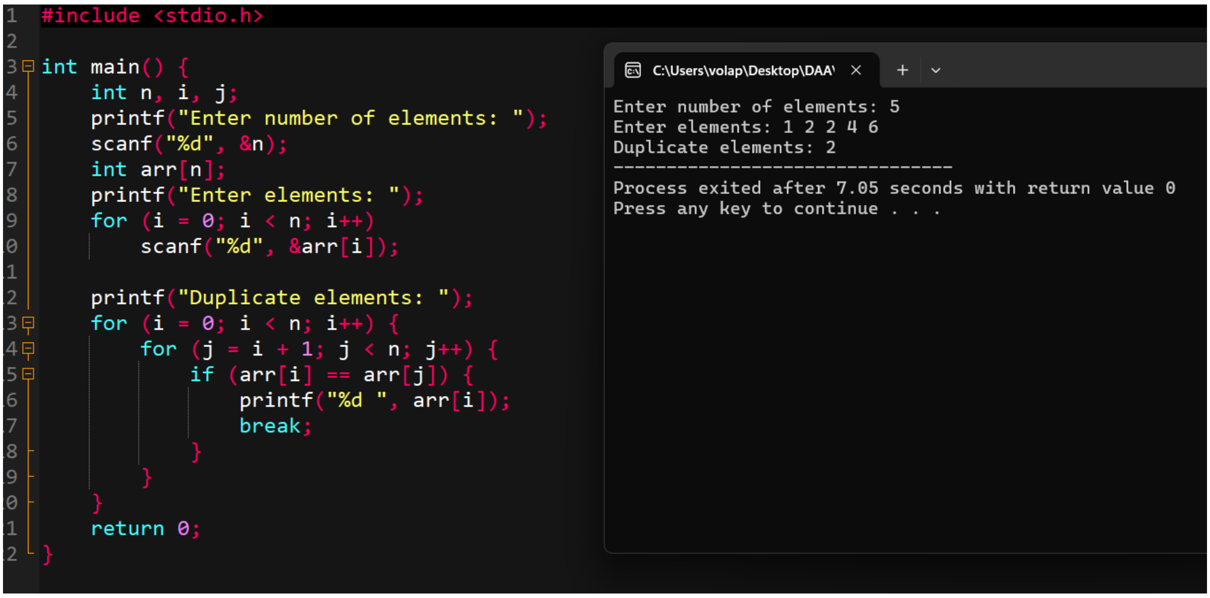
To write a C program to find duplicate elements in an array.

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: 1 2 2 4 6

Output: Duplicate: 2



```c
#include <stdio.h>

int main() {
    int n, i, j;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter elements: ");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Duplicate elements: ");
    for (i = 0; i < n; i++) {
        for (j = i + 1; j < n; j++) {
            if (arr[i] == arr[j]) {
                printf("%d ", arr[i]);
                break;
            }
        }
    }
    return 0;
}
```

```
Enter number of elements: 5
Enter elements: 1 2 2 4 6
Duplicate elements: 2
--------------------------------
Process exited after 7.05 seconds with return value 0
Press any key to continue . . .
```

3. Bigger Number in a Series.

Aim:

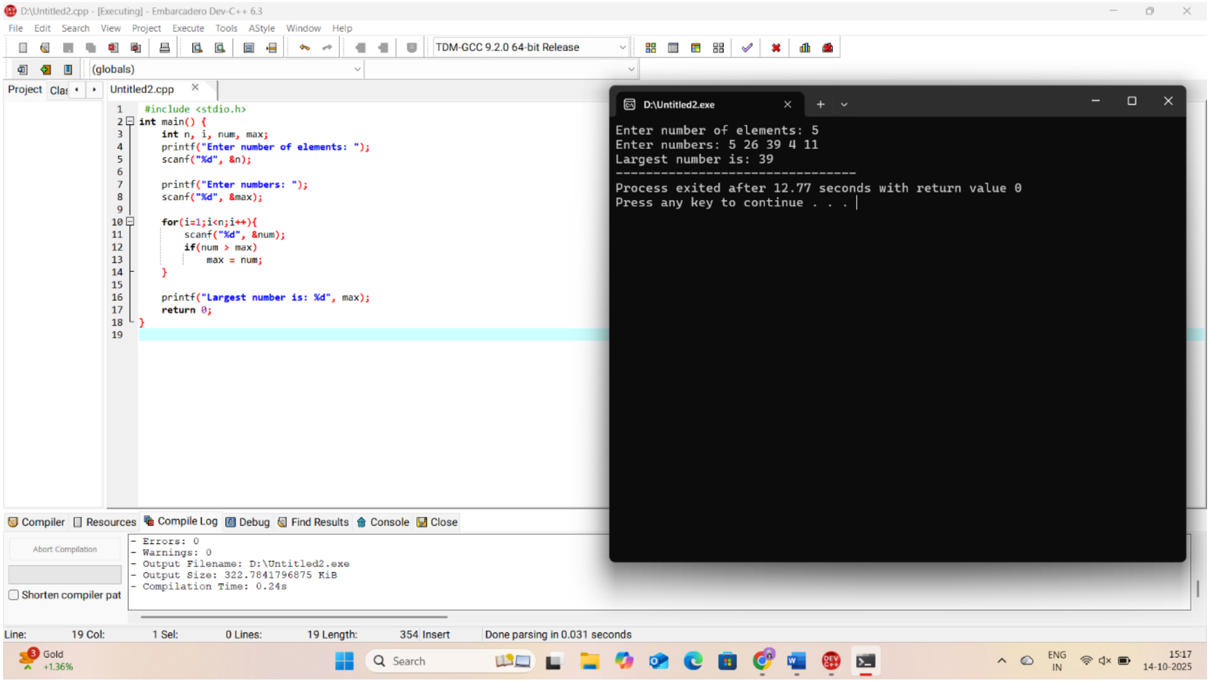
To write a C program to find the largest number from given numbers.

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



D:\Untitled2.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
Untitled2.cpp
D:\Untitled2.exe
Compiler Resources Compile Log Debug Find Results Console Close
Abort Compilation
Shorten compiler pat
Gold +1.36%
Search
ENG IN
15:17 14-10-2025

```c
 #include <stdio.h>
int main() {
    int n, i, num, max;
    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter numbers: ");
    scanf("%d", &max);

    for(i=1;i<n;i++){
        scanf("%d", &num);
        if(num > max)
            max = num;
    }

    printf("Largest number is: %d", max);
    return 0;
}
```

```
Enter number of elements: 5
Enter numbers: 5 26 39 4 11
Largest number is: 39
--------------------------------
Process exited after 12.77 seconds with return value 0
Press any key to continue . . .
```

```
- Errors: 0
- Warnings: 0
- Output Filename: D:\Untitled2.exe
- Output Size: 322.7841796875 KiB
- Compilation Time: 0.24s
```

Line: 19 Col: 1 Sel: 0 Lines: 19 Length: 354 Insert Done parsing in 0.031 seconds

4. Recursion – Factorial of a Given Number.

Aim:

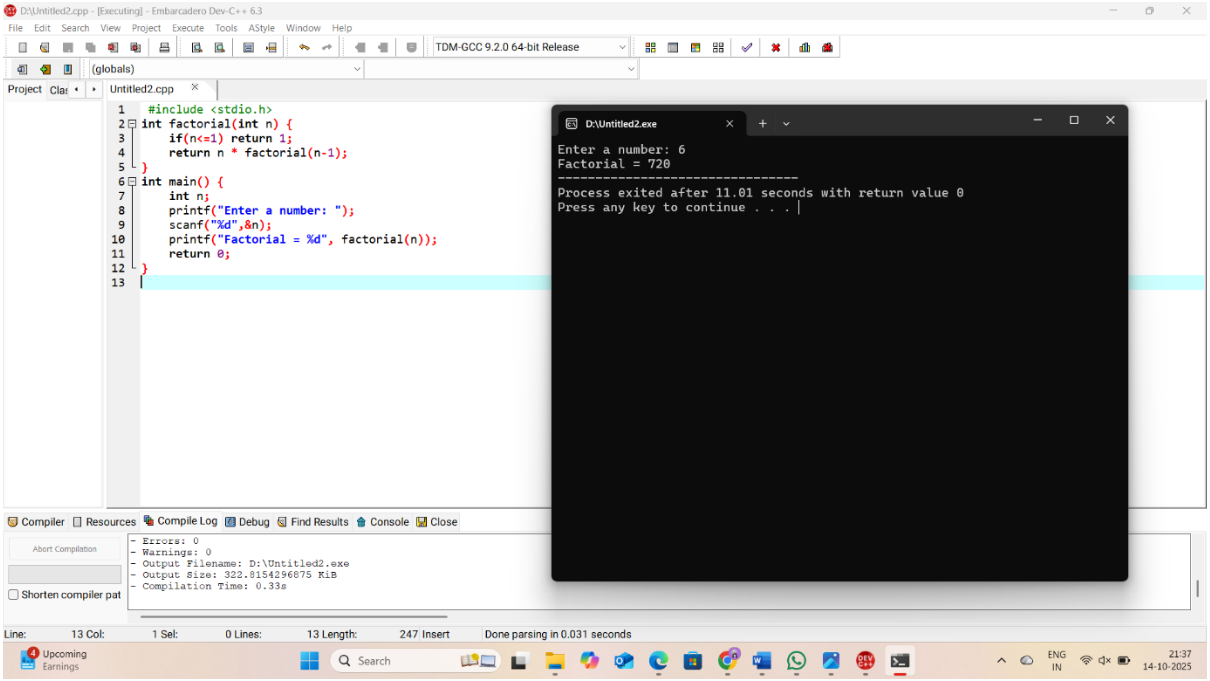
To write a C program to find the factorial of a number using recursion.

Algorithm:

1. Start
2. Read a number n
3. If n==0 or n==1 → return 1
4. Else return n * factorial(n-1)
5. Print result
6. Stop

Input: 6

Output: 720



D:\Untitled2.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
Untitled2.cpp
D:\Untitled2.exe
Compiler Resources Compile Log Debug Find Results Console Close
Abort Compilation
□ Shorten compiler pat
Line: 13 Col: 1 Sel: 0 Lines: 13 Length: 247 Insert Done parsing in 0.031 seconds
Upcoming Earnings
Search
ENG IN
21:37 14-10-2025

```c
 #include <stdio.h>
int factorial(int n) {
    if(n<=1) return 1;
    return n * factorial(n-1);
}
int main() {
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    printf("Factorial = %d", factorial(n));
    return 0;
}
```

```
Enter a number: 6
Factorial = 720
--------------------------------
Process exited after 11.01 seconds with return value 0
Press any key to continue . . .
```

```
- Errors: 0
- Warnings: 0
- Output Filename: D:\Untitled2.exe
- Output Size: 322.8154296875 KiB
- Compilation Time: 0.33s
```

5. Fibonacci Series.

Aim:

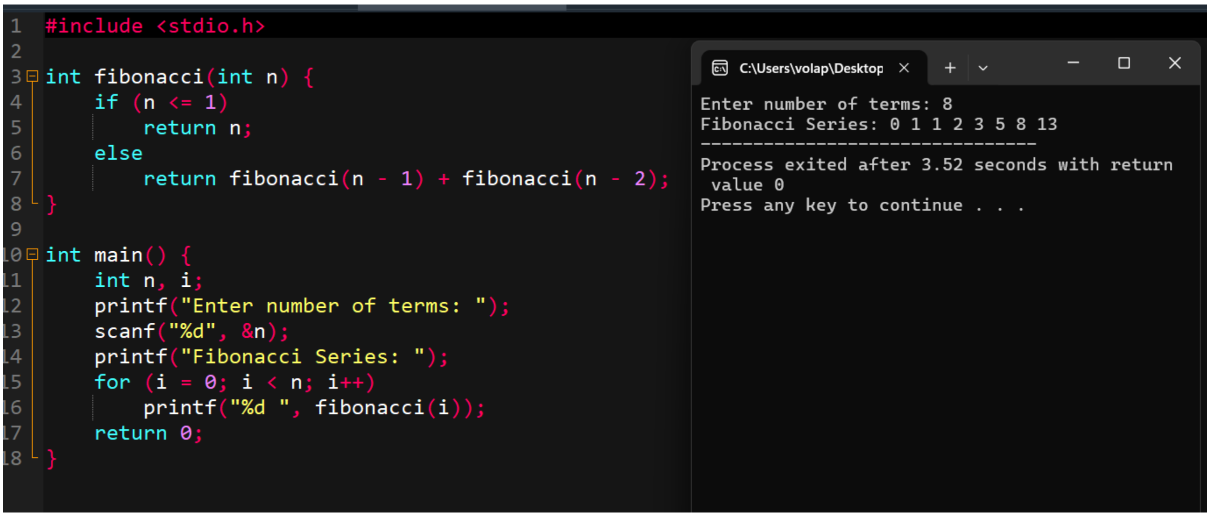
To write a C program to generate the Fibonacci series.

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: 8

Output: 0 1 1 2 3 5 8 13



```c
#include <stdio.h>

int fibonacci(int n) {
    if (n <= 1)
        return n;
    else
        return fibonacci(n - 1) + fibonacci(n - 2);
}

int main() {
    int n, i;
    printf("Enter number of terms: ");
    scanf("%d", &n);
    printf("Fibonacci Series: ");
    for (i = 0; i < n; i++)
        printf("%d ", fibonacci(i));
    return 0;
}
```

```
Enter number of terms: 8
Fibonacci Series: 0 1 1 2 3 5 8 13
--------------------------------
Process exited after 3.52 seconds with return
 value 0
Press any key to continue . . .
```

6. Two Order Homogeneous Recursion.

Aim:

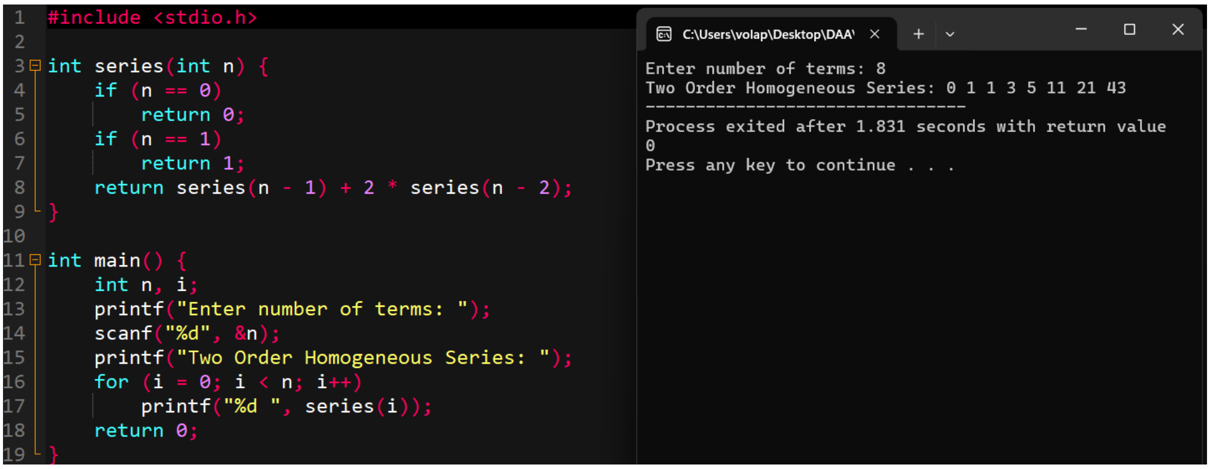
To write a C program using recursion for a second-order homogeneous recurrence relation.

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 3 5 11 21 43



```c
#include <stdio.h>

int series(int n) {
    if (n == 0)
        return 0;
    if (n == 1)
        return 1;
    return series(n - 1) + 2 * series(n - 2);
}

int main() {
    int n, i;
    printf("Enter number of terms: ");
    scanf("%d", &n);
    printf("Two Order Homogeneous Series: ");
    for (i = 0; i < n; i++)
        printf("%d ", series(i));
    return 0;
}
```

```
Enter number of terms: 8
Two Order Homogeneous Series: 0 1 1 3 5 11 21 43
--------------------------------
Process exited after 1.831 seconds with return value
0
Press any key to continue . . .
```

7. Leap Year

Aim:

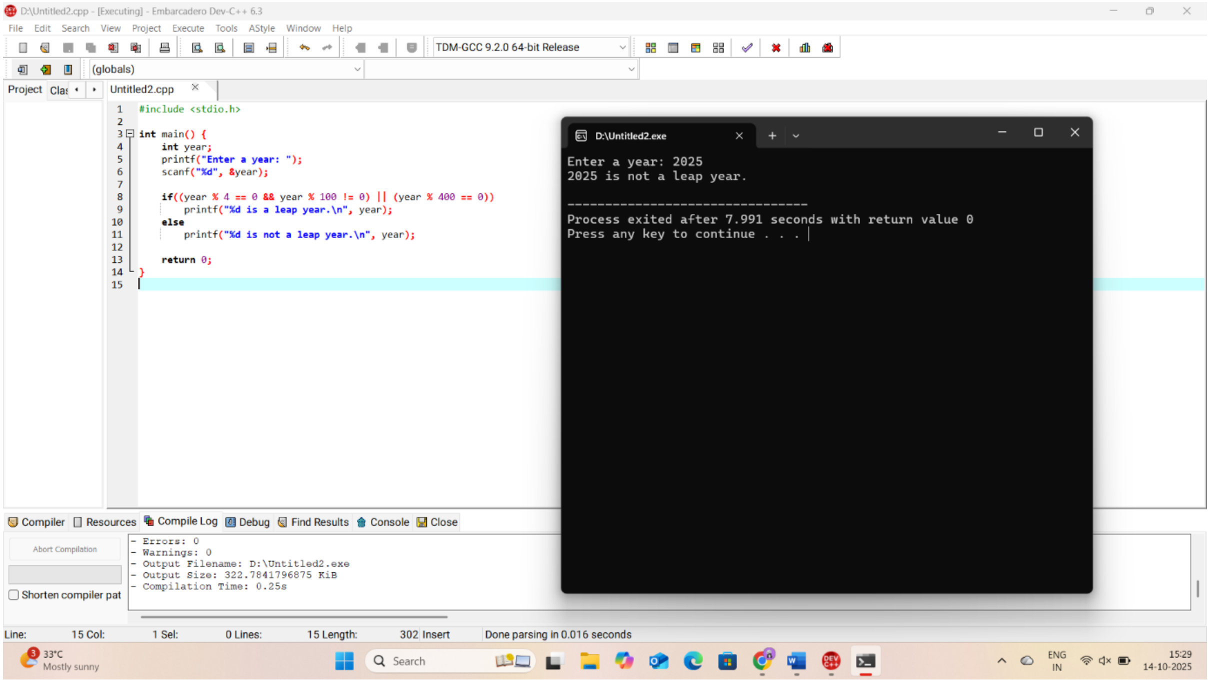
To write a C program to check whether a year is a leap year.

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



D:\Untitled2.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
Untitled2.cpp
D:\Untitled2.exe
Compiler Resources Compile Log Debug Find Results Console Close
Abort Compilation
Shorten compiler pat
Line: 15 Col: 1 Sel: 0 Lines: 15 Length: 302 Insert Done parsing in 0.016 seconds
33°C Mostly sunny
Search
ENG IN
15:29 14-10-2025

```c
#include <stdio.h>

int main() {
    int year;
    printf("Enter a year: ");
    scanf("%d", &year);

    if((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
        printf("%d is a leap year.\n", year);
    else
        printf("%d is not a leap year.\n", year);

    return 0;
}
```

```
Enter a year: 2025
2025 is not a leap year.

--------------------------------
Process exited after 7.991 seconds with return value 0
Press any key to continue . . .
```

```
- Errors: 0
- Warnings: 0
- Output Filename: D:\Untitled2.exe
- Output Size: 322.7841796875 KiB
- Compilation Time: 0.25s
```

8. Swapping of Numbers.

Aim:

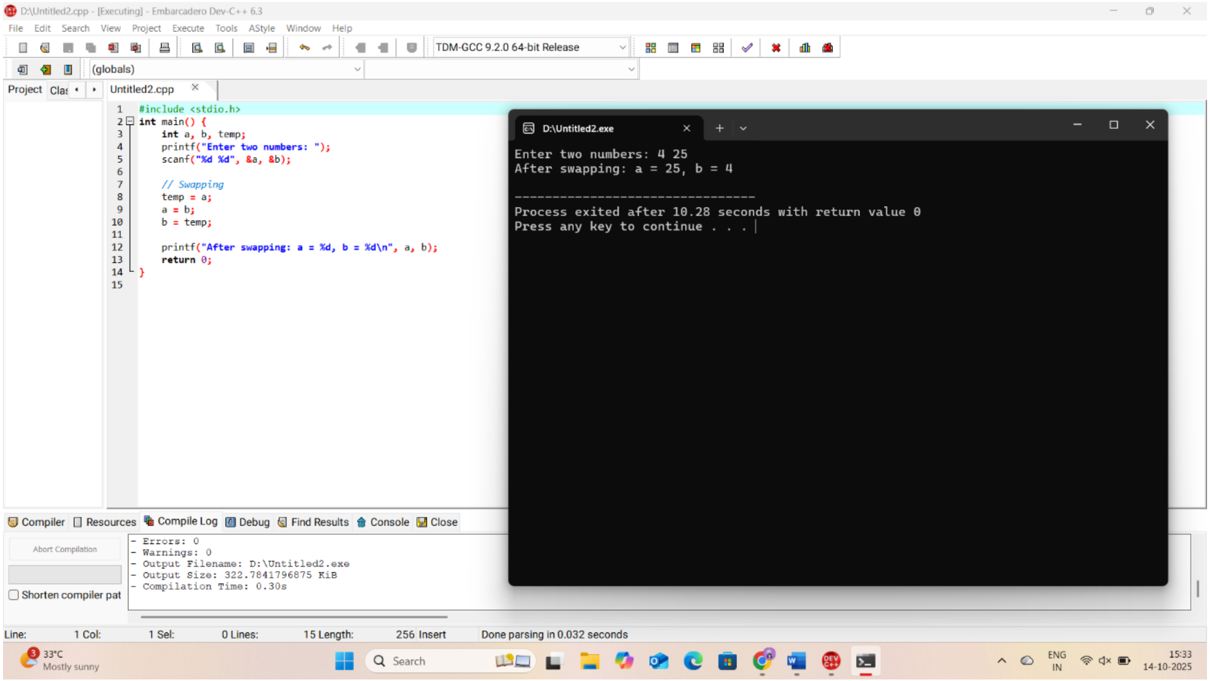
To write a C program to swap two numbers.

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



D:\Untitled2.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
Untitled2.cpp
D:\Untitled2.exe
Compiler Resources Compile Log Debug Find Results Console Close
Abort Compilation
□ Shorten compiler pat
33°C
Mostly sunny
Search
ENG
IN
15:33
14-10-2025

```c
#include <stdio.h>
int main() {
    int a, b, temp;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);

    // Swapping
    temp = a;
    a = b;
    b = temp;

    printf("After swapping: a = %d, b = %d\n", a, b);
    return 0;
}
```

```
Enter two numbers: 4 25
After swapping: a = 25, b = 4

--------------------------------
Process exited after 10.28 seconds with return value 0
Press any key to continue . . .
```

```
- Errors: 0
- Warnings: 0
- Output Filename: D:\Untitled2.exe
- Output Size: 322.7841796875 KiB
- Compilation Time: 0.30s
```

```
Line:     1 Col:     1 Sel:     0 Lines:     15 Length:     256 Insert     Done parsing in 0.032 seconds
```

9. Identifying Palindrome

Aim:

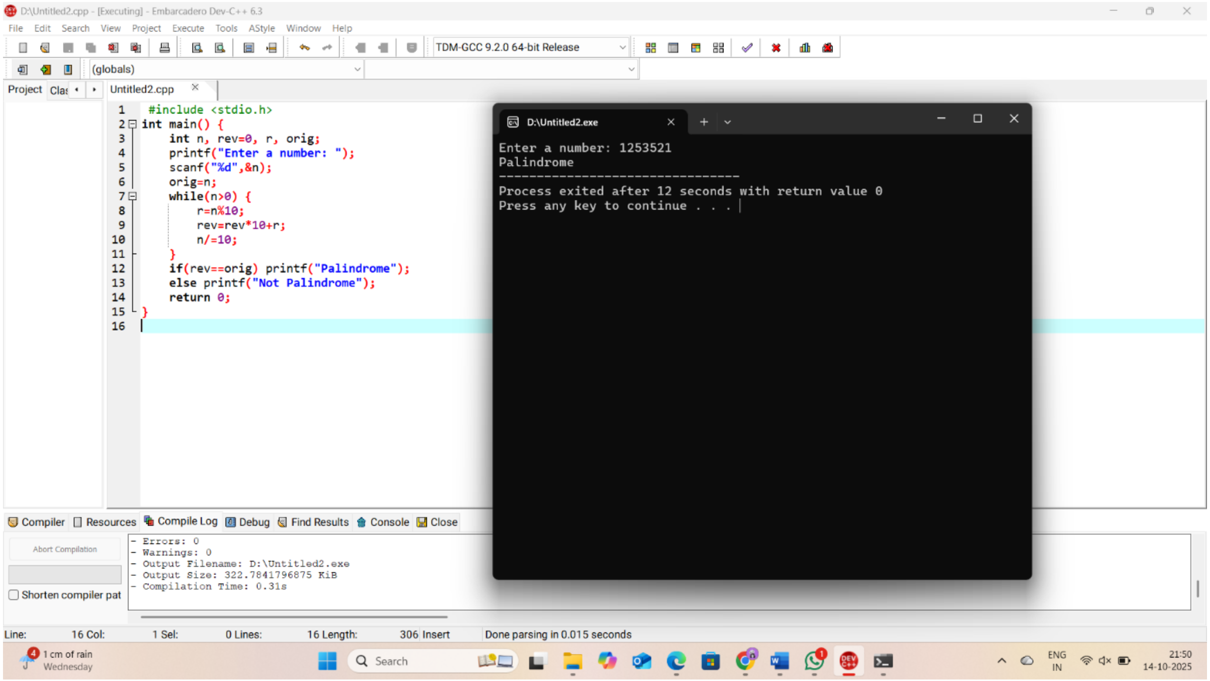
To write a C program to check whether a number is a palindrome.

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


D:\Untitled2.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
Untitled2.cpp
D:\Untitled2.exe
Compiler Resources Compile Log Debug Find Results Console Close
Abort Compilation
Shorten compiler pat
Line: 16 Col: 1 Sel: 0 Lines: 16 Length: 306 Insert Done parsing in 0.015 seconds
1 cm of rain
Wednesday
Search
ENG
IN
21:50
14-10-2025

```c
 #include <stdio.h>
int main() {
    int n, rev=0, r, orig;
    printf("Enter a number: ");
    scanf("%d",&n);
    orig=n;
    while(n>0) {
        r=n%10;
        rev=rev*10+r;
        n/=10;
    }
    if(rev==orig) printf("Palindrome");
    else printf("Not Palindrome");
    return 0;
}
```

```
Enter a number: 1253521
Palindrome
--------------------------------
Process exited after 12 seconds with return value 0
Press any key to continue . . .
```

```
- Errors: 0
- Warnings: 0
- Output Filename: D:\Untitled2.exe
- Output Size: 322.7841796875 KiB
- Compilation Time: 0.31s
```

10. Prime Number

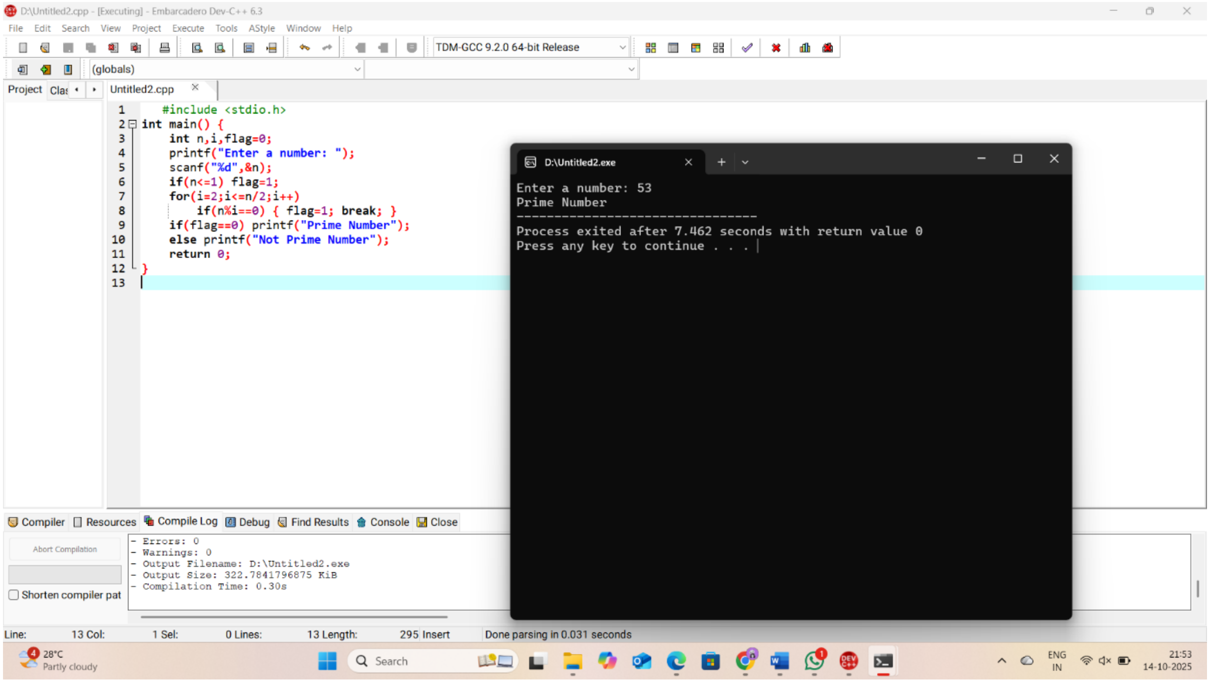
Aim:
To write a C program to check whether a number is prime.

Algorithm:

1. Start
2. Read n
3. If n<=1 → not prime
4. Check divisibility from 2 to n/2
5. If divisible → not prime
6. Else → prime
7. Stop

Input: 53

Output: prime number



D:\Untitled2.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
Untitled2.cpp
D:\Untitled2.exe
Compiler Resources Compile Log Debug Find Results Console Close
Abort Compilation
Shorten compiler pat
Line: 13 Col: 1 Sel: 0 Lines: 13 Length: 295 Insert Done parsing in 0.031 seconds
28°C Partly cloudy
Search
ENG IN
21:53 14-10-2025

```c
   #include <stdio.h>
int main() {
    int n,i,flag=0;
    printf("Enter a number: ");
    scanf("%d",&n);
    if(n<=1) flag=1;
    for(i=2;i<=n/2;i++)
        if(n%i==0) { flag=1; break; }
    if(flag==0) printf("Prime Number");
    else printf("Not Prime Number");
    return 0;
}
```

```
Enter a number: 53
Prime Number
--------------------------------
Process exited after 7.462 seconds with return value 0
Press any key to continue . . .
```

```
- Errors: 0
- Warnings: 0
- Output Filename: D:\Untitled2.exe
- Output Size: 322.7841796875 KiB
- Compilation Time: 0.30s
```