M.SRINIVASULU

192225089

1. An array is a data structure containing a collection of values or variables. The simplest type of array is a linear array or one-dimensional array. An array can be defined in C with the following syntax:

int Arr[5] = {12, 56, 34, 78, 100};

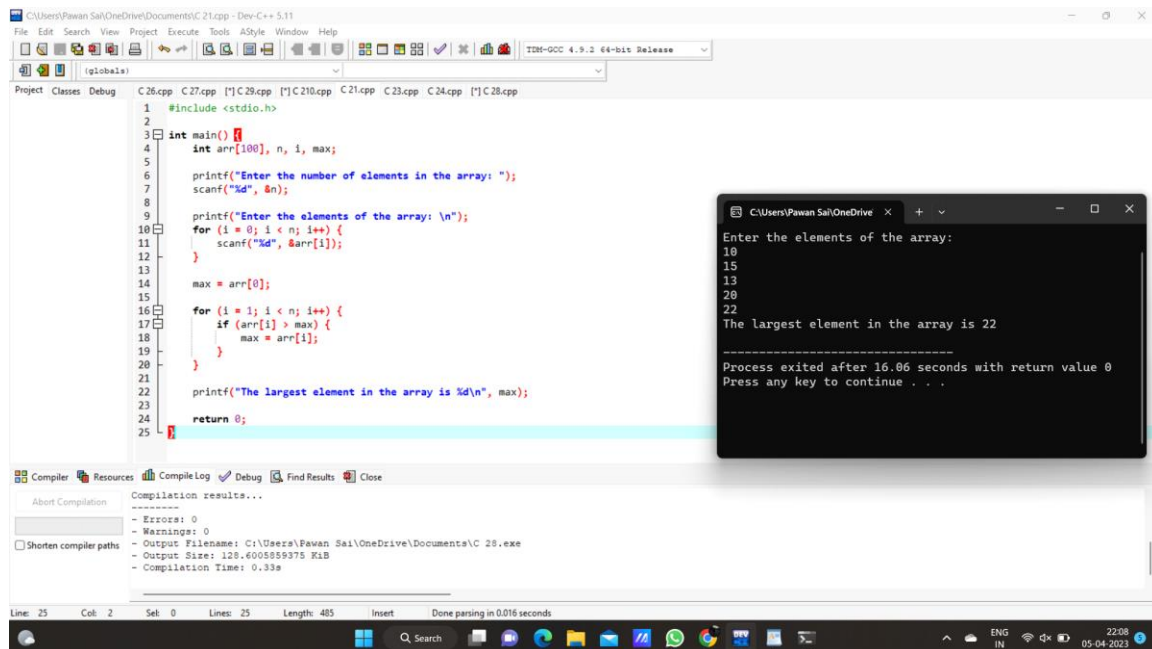/* here 12,56,34,78,100 are the elements at indices 0,1,2,3,4 respectively */

In this example, array Arr is a collection of 5 integers. Each integer can be identified and accessed by its index. The indices of the array start with 0, so the first element of the array will have index 0, the next will have index 1 and so on.

Largest element of the array is the array element which has the largest numerical value among all the array elements.

Examples:

If we are entering 5 elements (N = 5), with array element values as 12, 56, 34, 78 and 100

Then, largest element present in the given array is: 100

2.Problem Description

We have to write a program in C such that the program will read the elements of a one-dimensional array, then compares the elements and finds which are the largest two elements in a given array.

Expected Input and Output

1. Finding Largest 2 numbers in an array with unique elements:

If we are entering 5 elements (N = 5), with array element values as 2,4,5,8 and 7 then,

The FIRST LARGEST = 8

THE SECOND LARGEST = 7

2. Finding Largest 2 numbers in an array with recurring elements:

If we are entering 6 elements (N = 6), with array element values as 2,1,1,2,1 and 2 then,

The FIRST LARGEST = 2

THE SECOND LARGEST = 1

3. C Program finds second largest & smallest elements in an Array.


```c
# include <stdio.h>
int  main( )
{
    int  a[20], b[20], n, sml=0, pos, i, j, temp ;
    printf(" Enter the Numbers of terms: ") ;
    scanf("%d ",& n) ;
    printf("\n Enter the terms : \n") ;
    for ( i = 1 ; i <= n ; i++)
    {
    scanf("%d ",& a[i]) ;
    b[i] = a[i] ;
    }
    for ( i = 1 ; i <= n ; i++)
    {
    for ( j = 1 ; j <= n ; j++)
    {
    if ( a[i] <= a[j] )
    {
    temp = a[i] ;
    a[i] = a[j] ;
    a[j] = temp ;
    }}}
    printf("\n The Array Elements are: \n") ;
    for ( i = 1 ; i <= n ; i++)
    printf(" %d \t",b[i]) ;

    printf("\n Second Smallest Element is : %d",a[2]) ;
    printf("\n Second Largest Element is : %d",a[n-1]) ;
    return ( 0 ) ;
}
```

```
Enter the Numbers of terms: 5
5

Enter the terms :
16
14
12
18
19

The Array Elements are:
5       16      14      12      18
Second Smallest Element is : 12
Second Largest Element is : 16
------------------------------------
Process exited after 17.44 seconds with return value 0
Press any key to continue . . .
```

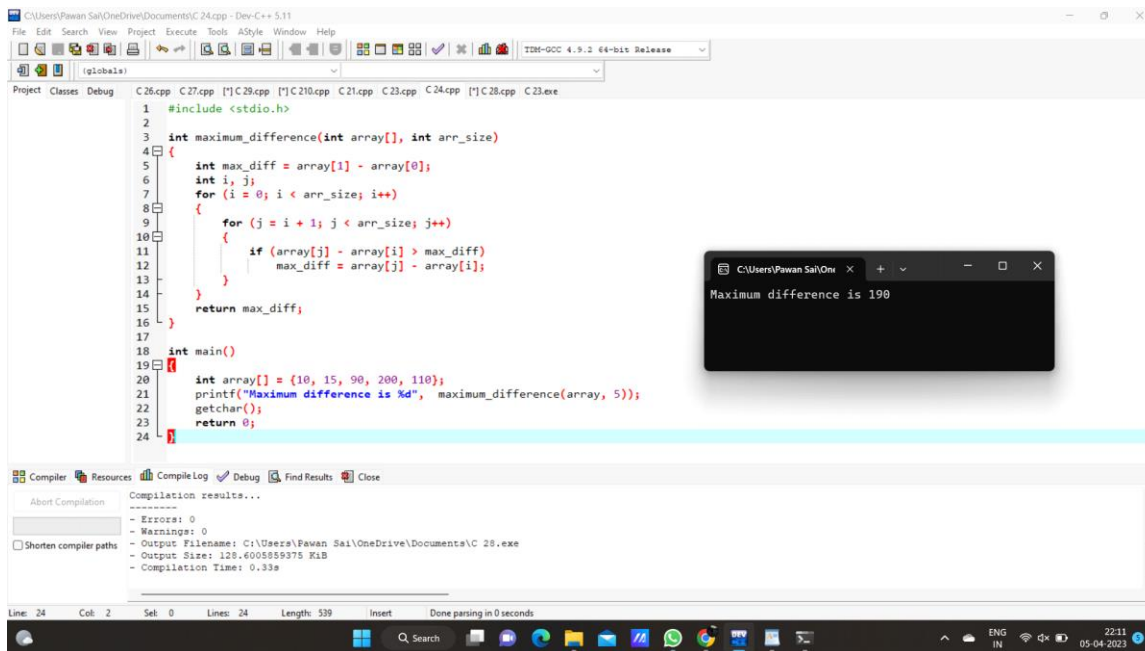4. C Program To Find Maximum Difference Between Two Elements in an Array

Example:

Consider the Following Array

int array[] = {10, 15, 90, 200, 110};

Output:

Maximum difference is 190

That is 200-10=190

5. **C program to remove duplicate elements in an Array?**

An **array** is a collection of similar data elements stored in a contiguous memory location.

Example: arr[5] = {2,7,1,23,5}
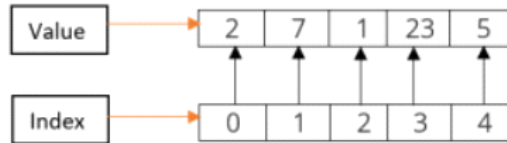
Example:

Input Array: 1,2,4,5,4,2,7,5
Output: Resultant Array after removing duplicates: 1,2,4,5,7

6. C Program to put even & odd elements of an array in 2 separate arrays.


```c
#include <stdio.h>
int main()
{
    int arr[] = {2, 5, 6, 9, 10, 11};
    int n = sizeof(arr) / sizeof(arr[0]);
    int even_arr[n], odd_arr[n];
    int even_count = 0, odd_count = 0;
    for (int i = 0; i < n; i++) {
        if (arr[i] % 2 == 0) {
            even_arr[even_count] = arr[i];
            even_count++;
        } else {
            odd_arr[odd_count] = arr[i];
            odd_count++;
        }
    }
    printf("Even elements: ");
    for (int i = 0; i < even_count; i++) {
        printf("%d ", even_arr[i]);
    }
    printf("\nOdd elements: ");
    for (int i = 0; i < odd_count; i++) {
        printf("%d ", odd_arr[i]);
    }

    return 0;
}
```

Output:
```
Even elements: 2 6 10
Odd elements: 5 9 11
---------------------------------------
Process exited after 0.02357 seconds with return value 0
Press any key to continue . . .
```

7. Reversing an array means substituting the last element in the first position and vice

versa and doing such a thing for all elements of the array. For example, first element is

swapped with last, second element is swapped by second last and so on.

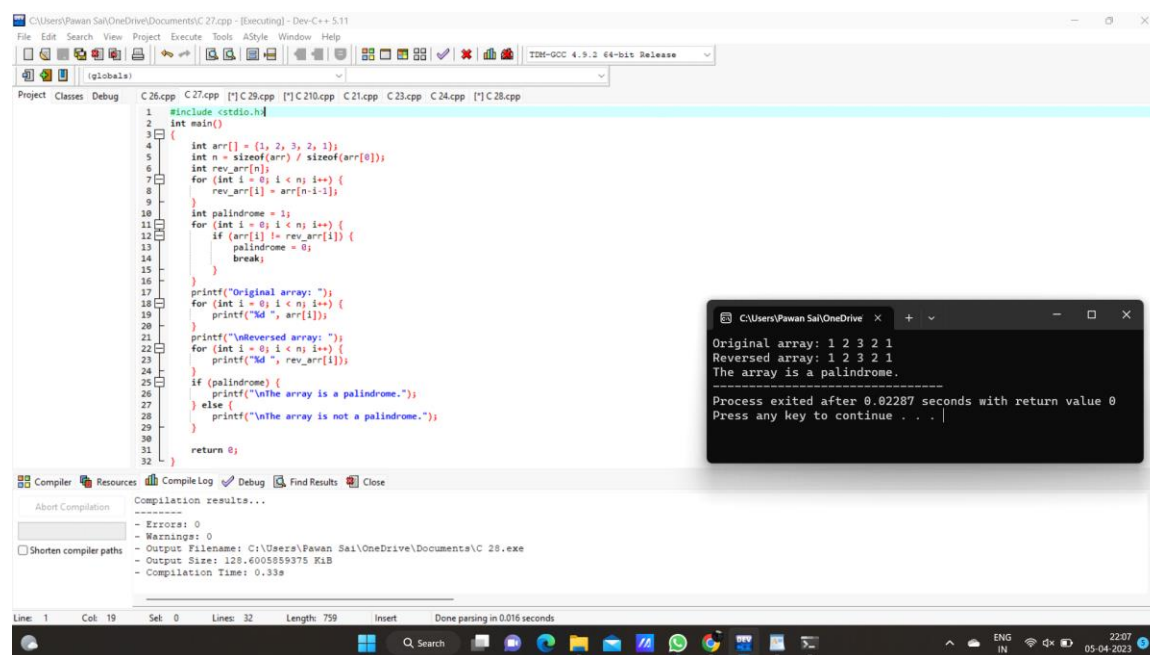Such arrays where the original and reversed arrays are equal are called palindrome

arrays.

Examples:

Input array: [1,2,3,4]

Reversed array: [4,3,2,1]

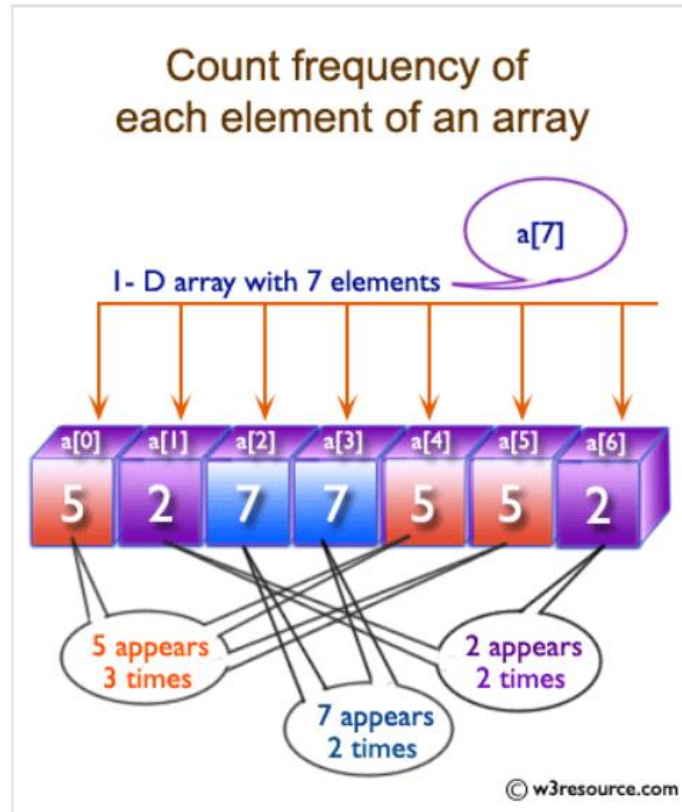Input array: [3,2,1]

Reversed array: [1,2,3]

8. Write a program in C to count the frequency of each element of an array.

9. C Program to sort an array in descending order.

Problem Description

This program will implement a one-dimensional array of some fixed size, filled with some

random numbers, then will sort all the filled elements of the array.

Enter the value of N

5

Enter the numbers

234

780

130

56

90

The numbers arranged in descending order are given below

780

234

130

90

56

```c
#include <stdio.h>
int main()
{
    int arr[100], n, i, j, temp;

    printf("Enter the value of N: ");
    scanf("%d", &n);

    printf("Enter the numbers: ");
    for(i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    for(i = 0; i < n; i++) {
        for(j = i+1; j < n; j++) {
```

```
Compiler  Resources  Compile Log  Debug  Find Results  Close
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Pawan Sai\OneDrive\Documents\C 28.exe
- Output Size: 128.6005859375 KiB
- Compilation Time: 0.33s
```

Console output:
```
Enter the value of N: 5
Enter the numbers: 15
22
13
12
24
The numbers arranged in descending order are given below:
24
22
15
13
12

------------------------------------
Process exited after 17.54 seconds with return value 0
Press any key to continue . . .
```

```c
        for(j = i+1; j < n; j++) {
            if(arr[i] < arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    printf("The numbers arranged in descending order are given below:\n");
    for(i = 0; i < n; i++) {
        printf("%d\n", arr[i]);
    }

    return 0;
```

```
Compiler  Resources  Compile Log  Debug  Find Results  Close
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Pawan Sai\OneDrive\Documents\C 28.exe
- Output Size: 128.6005859375 KiB
- Compilation Time: 0.33s
```

10. Given an array arr[] where each element represents the max number of steps that can be

made forward from that index. The task is to find the minimum number of jumps to reach the

end of the array starting from index 0. If the end isn't reachable, return -1.

Examples:

Input: arr[] = {1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9}

Output: 3 (1-> 3 -> 9 -> 9)

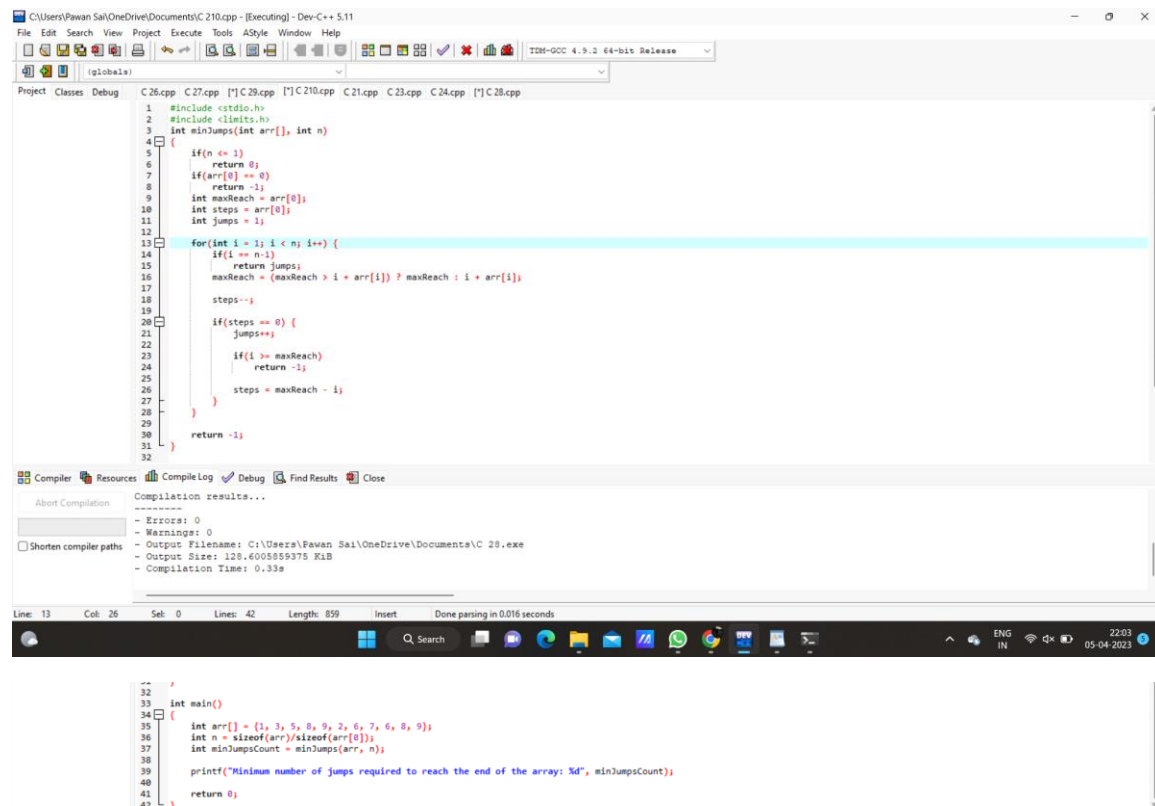Explanation: Jump from 1st element to 2nd element as there is only 1 step.

Now there are three options 5, 8 or 9. I

f 8 or 9 is chosen then the end node 9 can be reached. So 3 jumps are made.

Input: arr[] = {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}

Output: 10

Explanation: In every step a jump is needed so the count of jumps is 10.

```
Minimum number of jumps required to reach the end of the array: 3
--------------------------------
Process exited after 0.01813 seconds with return value 0
Press any key to continue . . .
```