**PRACTICAL NO.3**

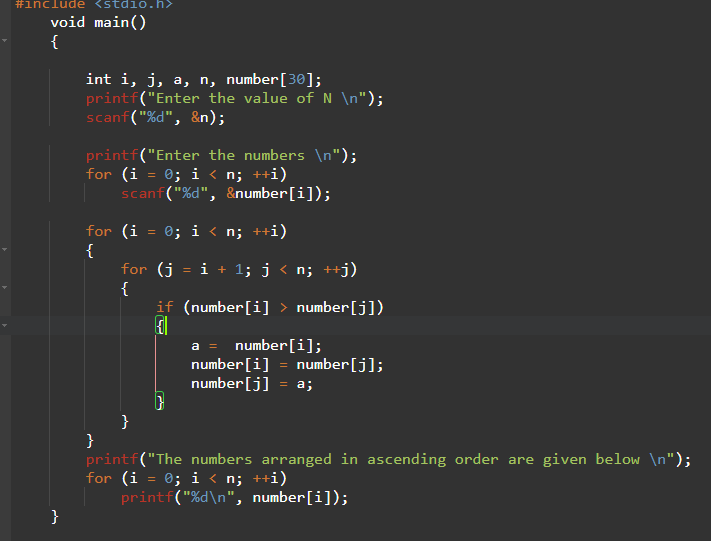
**ARRAYS:**

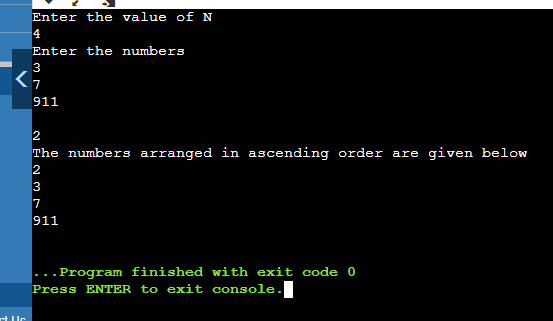
Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.To create an array, define the data type (like int) and specify the name of the array followed by **square brackets []**.

* Single dimensional array or 1-D array is the simplest form of arrays that can be found in C. This type of array consists of elements of similar types and these elements can be accessed through their indices.
* Syntax − type arrayName [ arraySize ];
* We initialize the array after the declaration by assigning the initial value to each element individually. We can use for loop, while loop, or do-while loop to assign the value to each element of the array.

**AIM:** one dimensional array

1.maximum of two numbers



****

**Max i m[i]**

**0      0  20**

**20   1  40**

**40    2 30**

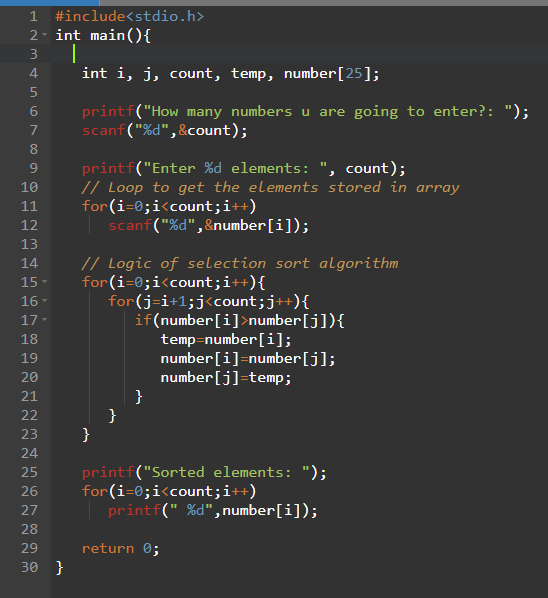
**40    3 60**

**60    4 90**

**90    5 80**

**max=90**

**2.selection sorting**

****

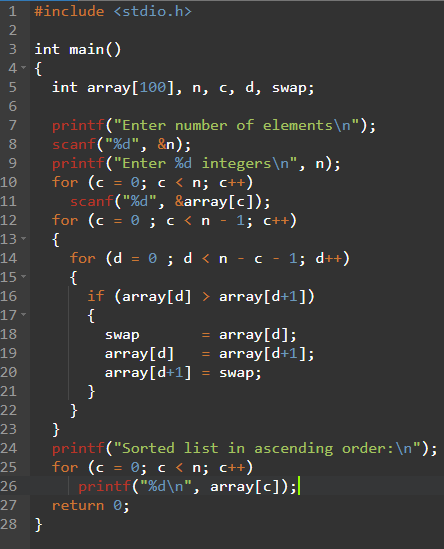
****

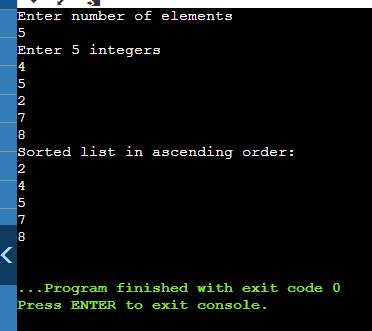
**Dry run**

|  |  |  |  |
| --- | --- | --- | --- |
| **4** | **3** | **2** | **1** |
| **a[0]** | **a[1]** | **a[2]** | **a[3]** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **i** | **j** | **a[j]** | **a[j+1]** | **array** |
| **0** | **0** | **4** | **3** | **3421** |
| **0** | **1** | **4** | **2** | **3241** |
| **0** | **2** | **4** | **1** | **3214** |
| **4-0-1=3** | **Exit j** |  |  |  |
| **1** | **0** | **3** | **2** | **2314** |
| **1** | **1** | **3** | **1** | **2134** |
| **4-1-1=2** | **Exit j** |  |  |  |
| **2** | **0** | **2** | **1** | **1234** |
| **4-2-1=1** | **Exit j** |  |  |  |

**3.bubble sorting**

****

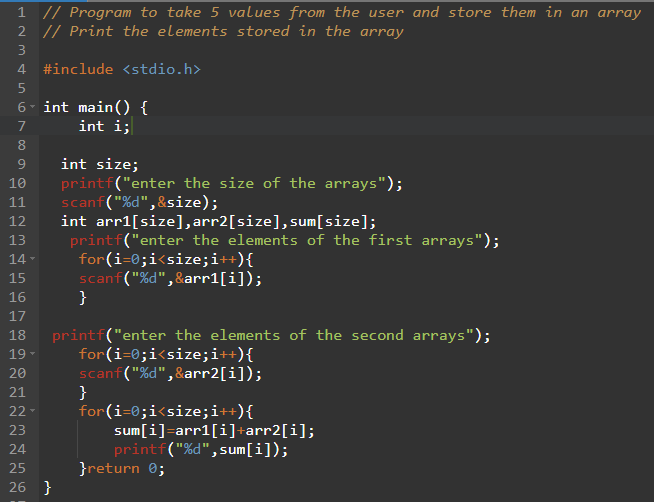
****

**Dry run:**

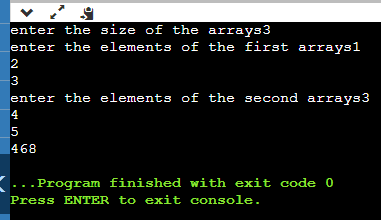
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **i** | **j** | **a[i]** | **a[j]** |  |
| **0** | **1** | **4** | **3** | **3 4 2 1** |
| **0** | **2** | **3** | **2** | **2 4 3 1** |
| **0** | **3** | **2** | **1** | **1 4 3 2** |
| **1** | **2** | **4** | **3** | **1 3 4 2** |
| **1** | **3** | **3** | **2** | **1 2 4 3** |
| **2** | **3** | **4** | **3** | **1 2 3 4** |
|  |  |  |  |  |

**1.** **Addition of two arrays**

**Coding:**



**Output:**

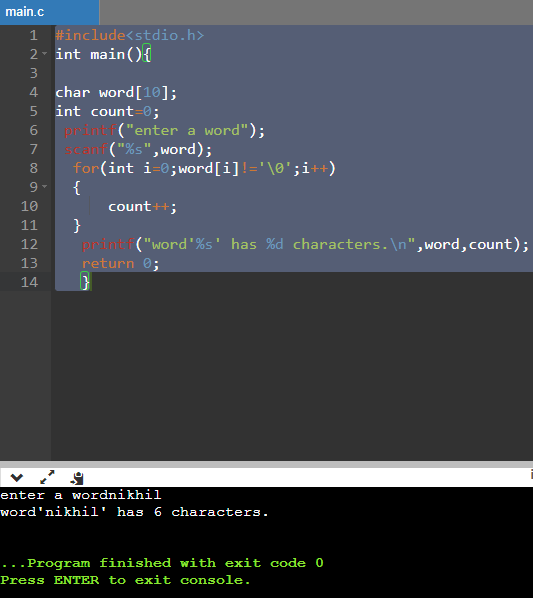
****

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 6 | 7 | 8 |
| 6 | 8 | 10 | 12 |

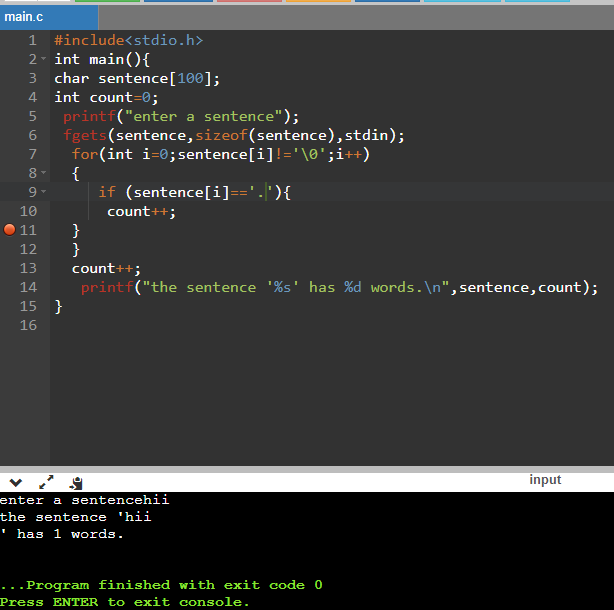
**3.** **Count characters of word**

**Coding:**

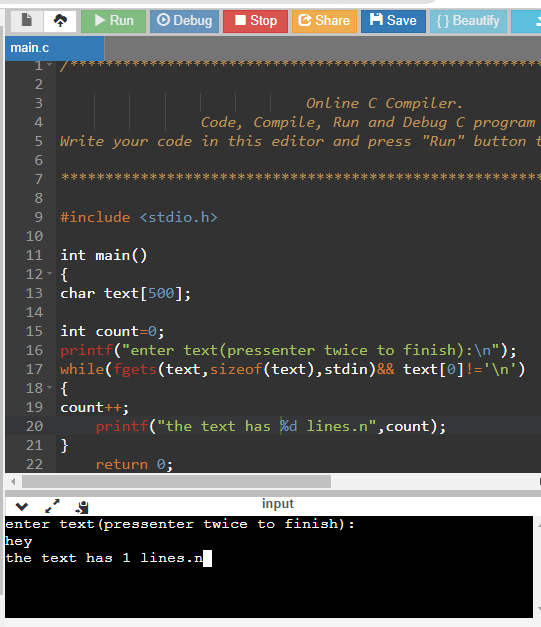
****

**4.** **Count number of words in sentence**

**Coding:**

****

**5.** **Count number of lines**

****

**Dry run**