ARRAYS

To handle large volume of data in terms of reading, processing and printing, a powerful data type is required that can facilitate efficient storing, accessing and manipulation of data items. C supports a derived data type known as array that can be used for such applications.

Definition

An array is a fixed-size sequenced collection of elements of the same data type.

Types of Arrays:

There are three types of arrays are there. They are

- Single –dimensional array (or) one dimensional array.
- Two-dimensional array.
- Multi dimensional array.

One-dimensional Arrays

Arrays must be declared before they are used so that the compiler can allocate space for them in memory.

Declaration

The general form of array declaration is

type variable_name[size];

The **type** specifies the data type of the element that will be contained in the array, such as int, float or char. size indicates the maximum number of elements that can be stored inside the array.

For example,

int group[10];   --→ group is an array of type integer

int → 2 bytes x 10 = 20 bytes

declares **group** as an array to contain a maximum of 10 integer constants. Here, subscripts 0 to 9 are valid.

group[0]	group[1]	group[2]	group[3]	group[4]	group[5]	group[6]	group[7]	group[9]

2000   2002   2004   2006   2008   ….

float mark[5]; --→ mark[0] to mark[4]

Here, we declared an array, mark, of floating-point type and size 5. Meaning, it can hold 5 floating-point values.

- Arrays have 0 as the first index not 1.
- If the size of an array is n, to access the last element, (n-1) index is used. In this example, mark[4]
- Suppose the starting address of mark[0] is 2120 d. Then, the next address, mark[1], will be 2124d, address of amark2] will be 2128 d and so on. It's because the size of a float is 4 bytes.

Initialization

The general form of array initialization is

type array_name[size] = {list of values};

The values in the list are separated by commas.

Example: int number[3] = {0,0,0}; will assign zero to each element

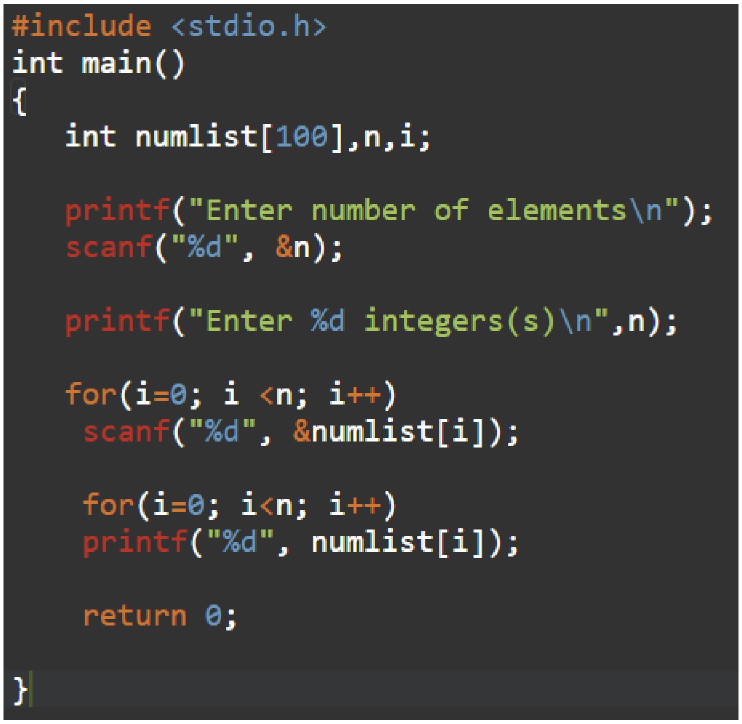
float total[5] = {0.0, 15.75, -10}; will initialize first three elements to 0.0, 15.75, -10 and the remaining two elements to zero.

Declare and initialize an integer array with 6 members 5,10,15,20,25,30 and print the position and value of each element

```
#include<stdio.h>

int main()

{
  int i;

  // declaring and Initialising array in C

  int value[6] = {5,10,15,20,25,30};

  for (i=0;i<6;i++)
  {
    // Accessing each variable using for loop
    printf("Position : [%d] , Value : %d \n", i, value[i]);

  }

  //Main Function return Statement
```

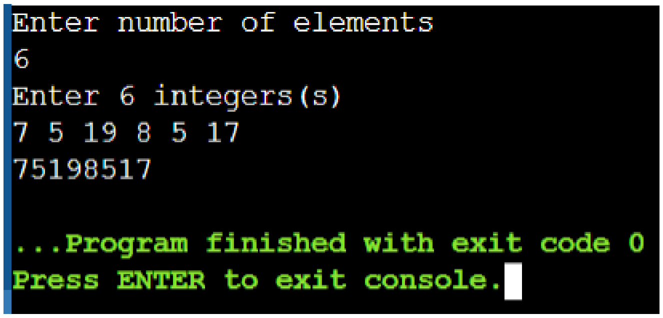
```
   return 0;
}
```

Program to input n numbers and print one-dimensional array



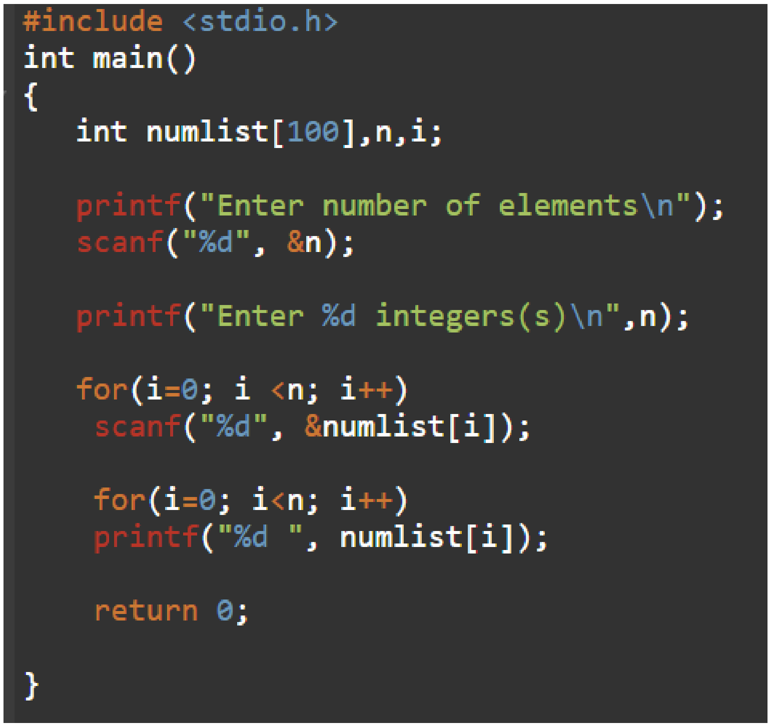
```
#include <stdio.h>
int main()
{
   int numlist[100],n,i;

   printf("Enter number of elements\n");
   scanf("%d", &n);

   printf("Enter %d integers(s)\n",n);

   for(i=0; i <n; i++)
    scanf("%d", &numlist[i]);

    for(i=0; i<n; i++)
    printf("%d", numlist[i]);

    return 0;

}
```

Output

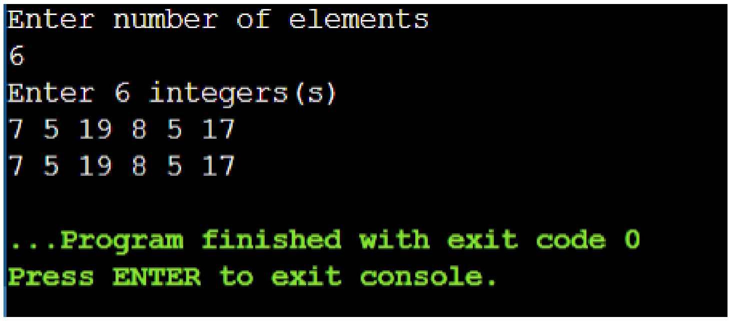


```
Enter number of elements
6
Enter 6 integers(s)
7 5 19 8 5 17
75198517

...Program finished with exit code 0
Press ENTER to exit console.
```

Program to print one-dimensional array with spaces between numbers



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

   printf("Enter number of elements\n");
   scanf("%d", &n);

   printf("Enter %d integers(s)\n",n);

   for(i=0; i <n; i++)
    scanf("%d", &numlist[i]);

    for(i=0; i<n; i++)
    printf("%d ", numlist[i]);

    return 0;

}
```

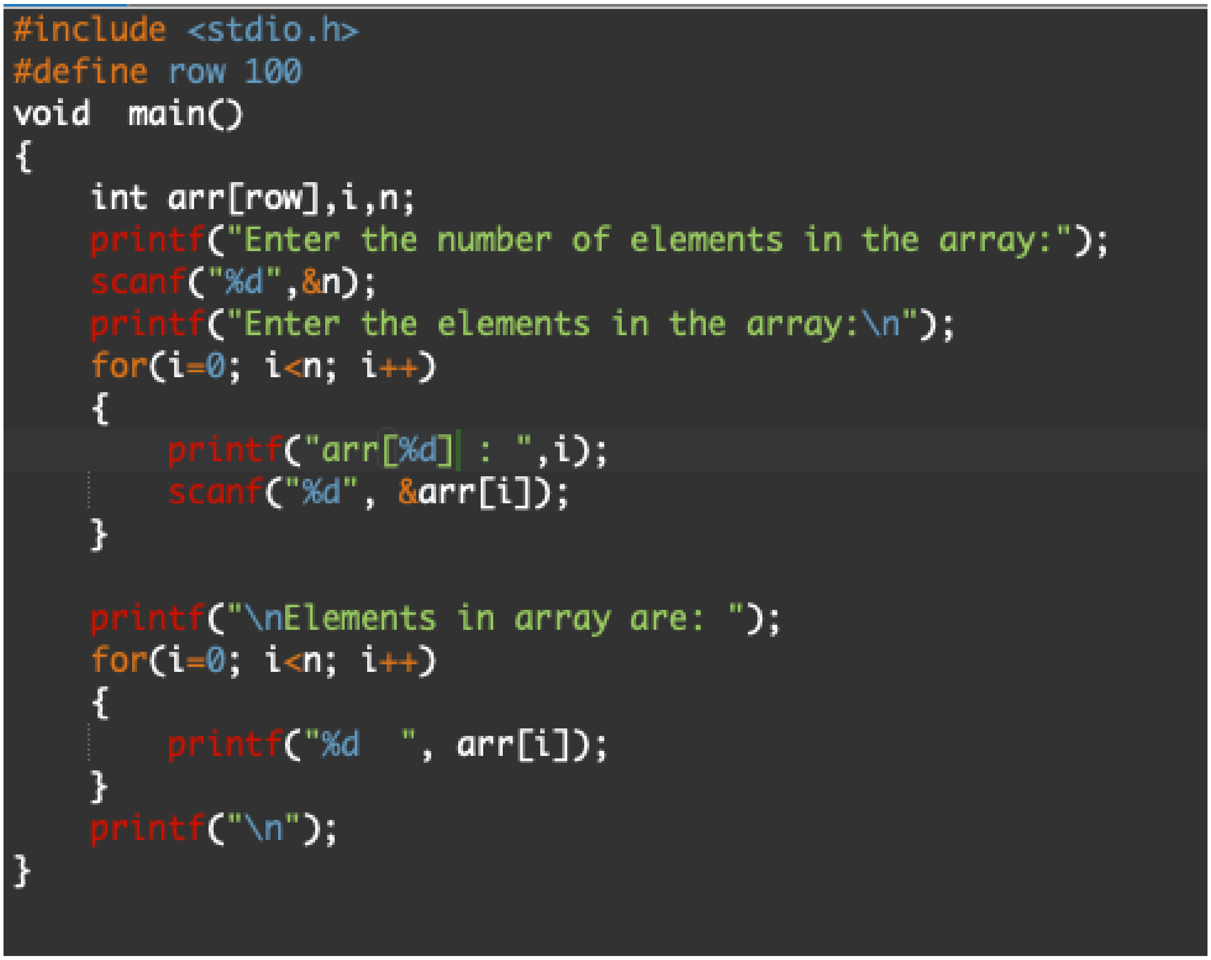


```
Enter number of elements
6
Enter 6 integers(s)
7 5 19 8 5 17
7 5 19 8 5 17

...Program finished with exit code 0
Press ENTER to exit console.
```

Program to print the size of arrays

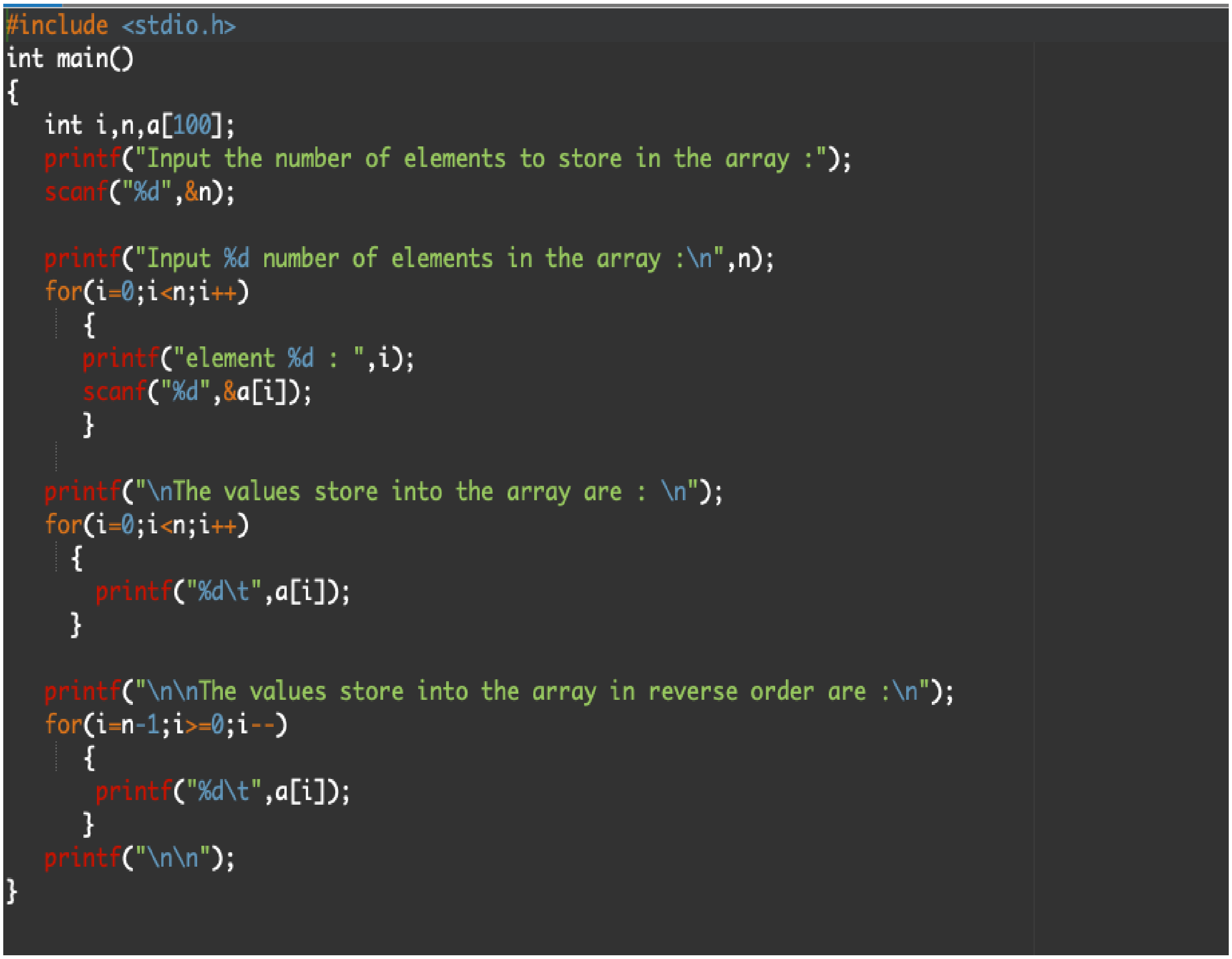
```c
#include <stdio.h>

int main() {


        // Declare Array Variable
    int array[5]= { 1, 2, 3, 4, 5 };

    int size;

    size = sizeof(array)/sizeof(int);

    printf("The Size of Array is %d\n", size);
    return 0;
}
```

Store elements in an array and print them

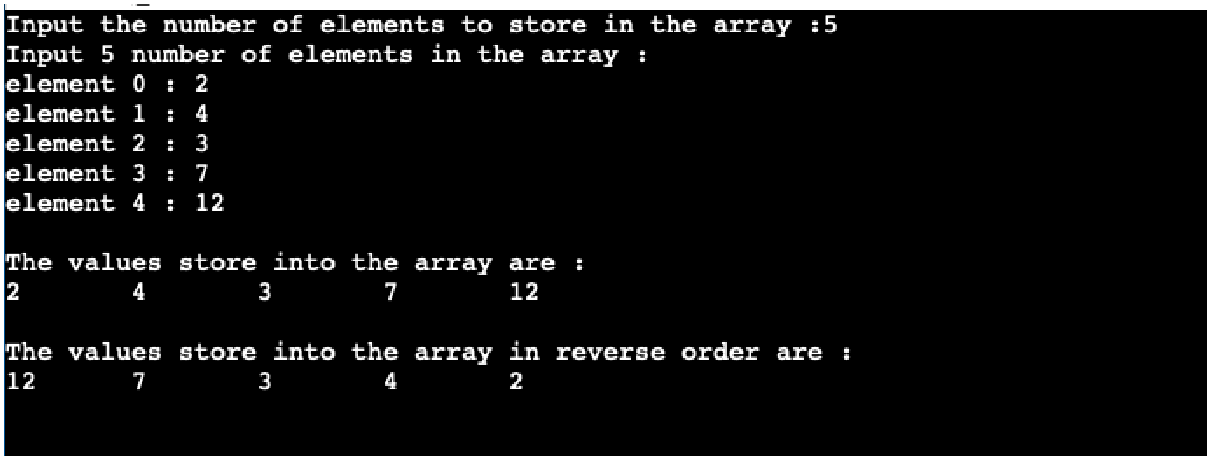


```c
#include <stdio.h>
#define row 100
void  main()
{
    int arr[row],i,n;
    printf("Enter the number of elements in the array:");
    scanf("%d",&n);
    printf("Enter the elements in the array:\n");
    for(i=0; i<n; i++)
    {
        printf("arr[%d] : ",i);
        scanf("%d", &arr[i]);
    }

    printf("\nElements in array are: ");
    for(i=0; i<n; i++)
    {
        printf("%d  ", arr[i]);
    }
    printf("\n");
}
```

To store elements in the array and print them in reverse order

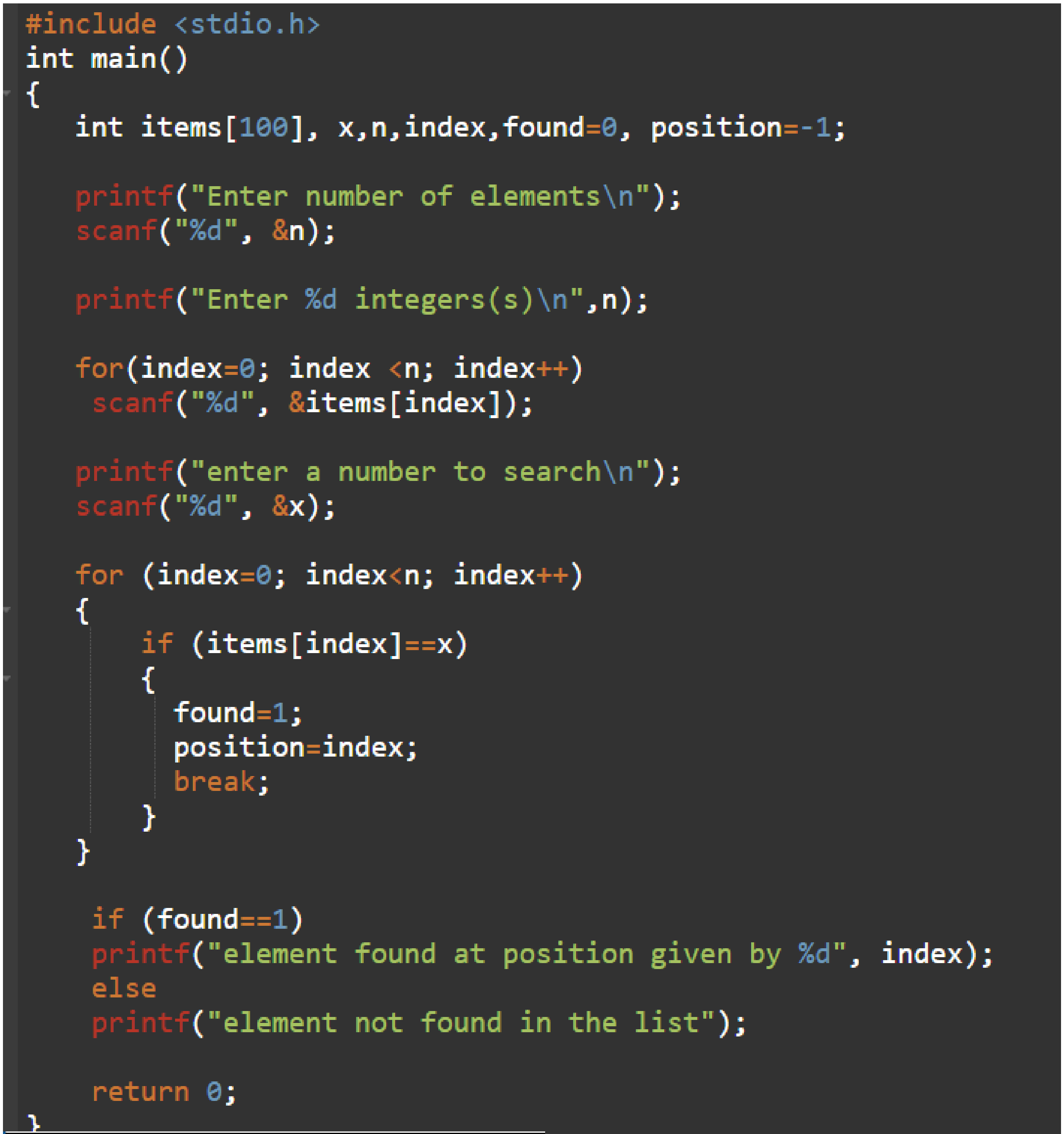


```
#include <stdio.h>
int main()
{
   int i,n,a[100];
   printf("Input the number of elements to store in the array :");
   scanf("%d",&n);

   printf("Input %d number of elements in the array :\n",n);
   for(i=0;i<n;i++)
      {
      printf("element %d : ",i);
      scanf("%d",&a[i]);
      }

   printf("\nThe values store into the array are : \n");
   for(i=0;i<n;i++)
     {
       printf("%d\t",a[i]);
     }

   printf("\n\nThe values store into the array in reverse order are :\n");
   for(i=n-1;i>=0;i--)
      {
       printf("%d\t",a[i]);
      }
   printf("\n\n");
}
```

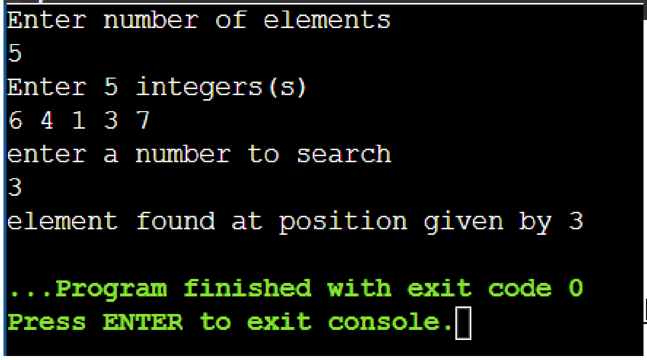


```
Input the number of elements to store in the array :5
Input 5 number of elements in the array :
element 0 : 2
element 1 : 4
element 2 : 3
element 3 : 7
element 4 : 12

The values store into the array are :
2       4       3       7       12

The values store into the array in reverse order are :
12      7       3       4       2
```

Program for linear search

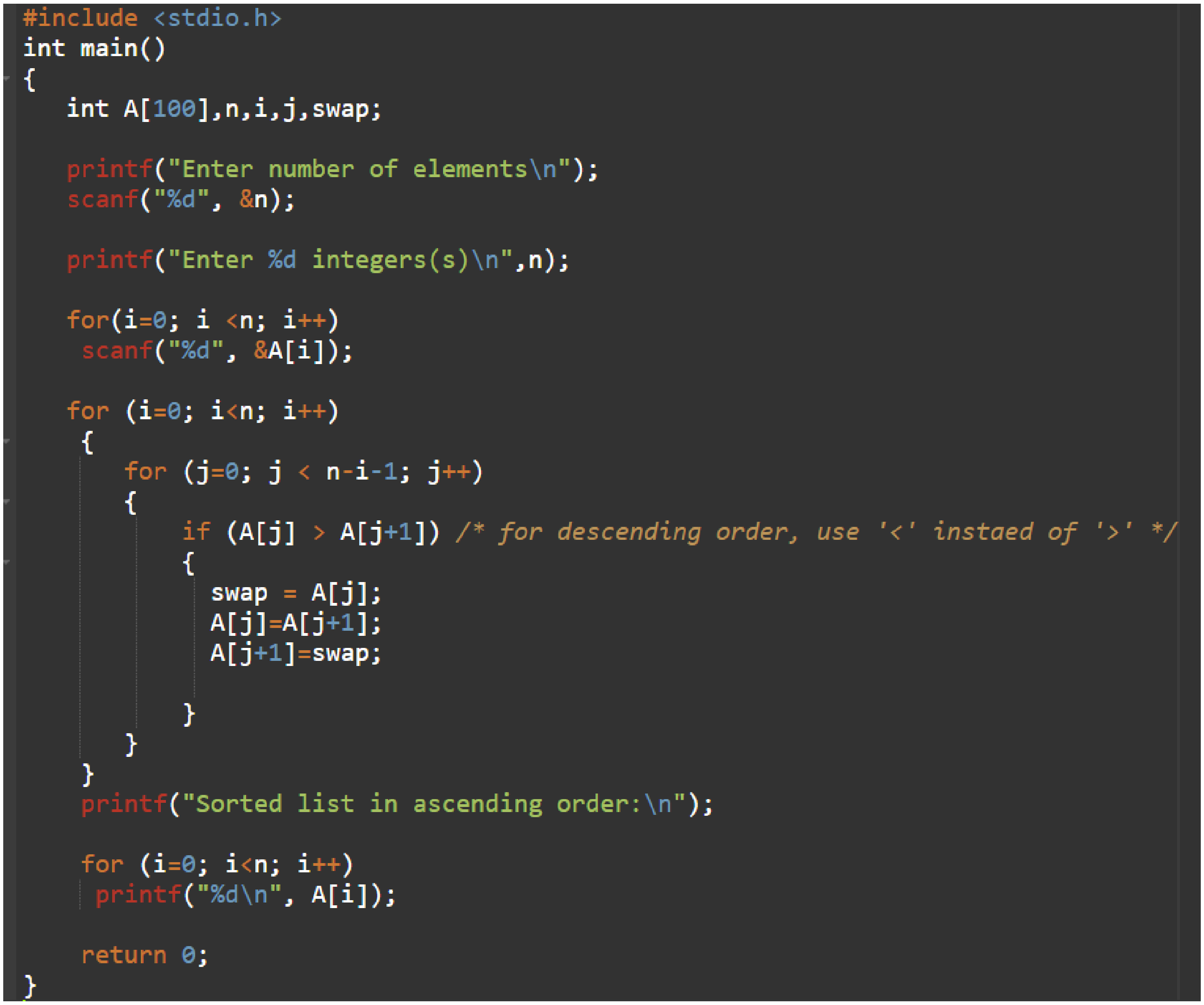


```c
#include <stdio.h>
int main()
{
   int items[100], x,n,index,found=0, position=-1;

   printf("Enter number of elements\n");
   scanf("%d", &n);

   printf("Enter %d integers(s)\n",n);

   for(index=0; index <n; index++)
    scanf("%d", &items[index]);

   printf("enter a number to search\n");
   scanf("%d", &x);

   for (index=0; index<n; index++)
   {
       if (items[index]==x)
       {
         found=1;
         position=index;
         break;
       }
   }

    if (found==1)
    printf("element found at position given by %d", index);
    else
    printf("element not found in the list");

    return 0;
}
```

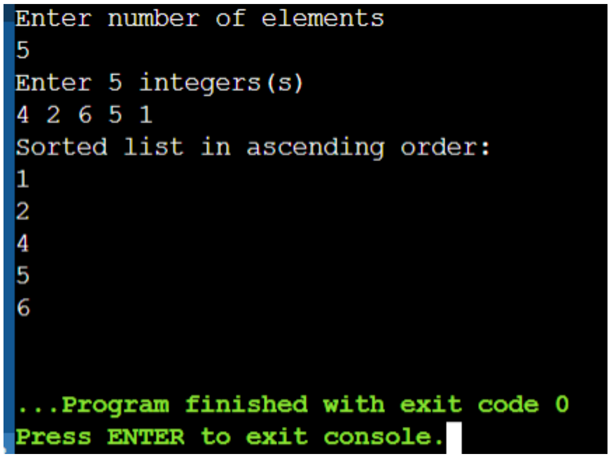


```
Enter number of elements
5
Enter 5 integers(s)
6 4 1 3 7
enter a number to search
3
element found at position given by 3

...Program finished with exit code 0
Press ENTER to exit console.
```

Program for bubble sort



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

   printf("Enter number of elements\n");
   scanf("%d", &n);

   printf("Enter %d integers(s)\n",n);

   for(i=0; i <n; i++)
    scanf("%d", &A[i]);

   for (i=0; i<n; i++)
    {
       for (j=0; j < n-i-1; j++)
       {
           if (A[j] > A[j+1]) /* for descending order, use '<' instaed of '>' */
           {
             swap = A[j];
             A[j]=A[j+1];
             A[j+1]=swap;

           }
       }
    }
    printf("Sorted list in ascending order:\n");

    for (i=0; i<n; i++)
     printf("%d\n", A[i]);

    return 0;
}
```



```
Enter number of elements
5
Enter 5 integers(s)
4 2 6 5 1
Sorted list in ascending order:
1
2
4
5
6


...Program finished with exit code 0
Press ENTER to exit console.
```