1. Write a C Program to find Largest Element of an Integer Array.  
   Here the number of elements in the array ‘n’ and the elements of the array is read from the test data.   
     
   Use the printf statement given below to print the largest element. printf("Largest element = %d", largest);

#include <stdio.h>

int main()

{

int i, n, largest;

int arr[100];

scanf("%d", &n); /\*Accepts total number of elements from the test data \*/

for(i = 0; i < n; ++i)

{

scanf("%d", &arr[i]); /\* Accepts the array element from test data \*/

}

largest=0;

for (i=0;i<n;i++)

{

if(arr[i]>largest)

largest=arr[i];

}

printf("Largest element = %d", largest);

}

1. Write a C Program to print the array elements in reverse order (Not reverse sorted order. Just the last element will become first element, second last element will become second element and so on)  
   Here the size of the array, ‘n’ and the array elements is accepted from the test case data.  
     
   The last part i.e. printing the array is also written. You have to complete the program so that it prints in the reverse order

#include<stdio.h>

int main() {

int arr[20], i, n;

scanf("%d", &n); /\* Accepts the number of elements in the array \*/

for (i = 0; i < n; i++)

scanf("%d", &arr[i]); /\*Accepts the elements of the array \*/

int j, temp;

j = i - 1; // last Element of the array

i = 0; // first element of the array

while (i < j) {

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

i++;

j--;

}

for (i = 0; i < n; i++) {

printf("%d\n", arr[i]); // For printing the array elements

}

return (0);

}

1. Write a C program to read Two One Dimensional Arrays of same data type (integer type) and merge them into another One Dimensional Array of same type.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Private Test cases used for evaluation** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| Test Case 1 | 3  15  45  25  3  60  70  80 | 15\n  45\n  25\n  60\n  70\n  80 | 15\n  45\n  25\n  60\n  70\n  80\n | Passed |
| Test Case 2 | 4  90  80  10  30  2  25  75 | 90\n  80\n  10\n  30\n  25\n  75 | 90\n  80\n  10\n  30\n  25\n  75\n | Passed |

#include<stdio.h>

int main()

{

int arr1[20], arr2[20], array\_new[40], n1, n2, size, i;

/\*n1 size of first array (i.e. arr1[]), n2 size of second array(i.e. arr2[]),

size is the total size of the new array (array\_new[]) \*/

scanf("%d", &n1); //Get the size of first array from test data and store it in n1.

for (i = 0; i < n1; i++)

scanf("%d", &arr1[i]); //Accepts the values for first array

scanf("%d", &n2); //Get the size of second array from test data and store it in n2.

for (i = 0; i < n2; i++)

scanf("%d", &arr2[i]); //Accepts the values for second array

size=n1+n2;

for(i=0;i<n1;i++)

array\_new[i]=arr1[i];

int j=0;

for(i=n1;i<size;i++)

{ array\_new[i]=arr2[j];

j++;

}

for (i = 0; i < size; i++) {

printf("%d\n", array\_new[i]);

}

}

Write a C Program to delete duplicate elements from an array of integers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Private Test cases used for evaluation** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| Test Case 1 | 6  50  6  7  7  2  7 | 50\n  6\n  7\n  2 | 50\n  6\n  7\n | Wrong Answer |
| Test Case 2 | 7  2  4  2  6  4  2  4 | 2\n  4\n  6 | 2\n | Wrong Answer |

#include<stdio.h>

int main()

{

int array[50], i, size;

scanf("%d", &size); /\*Accepts the size of array from test case data \*/

for (i = 0; i < size; i++)

scanf("%d", &array[i]); /\* Read the array elements from the test case data \*/

int j,k,count=0;

for(i=0;i<size;i++)

{ for(j=i+1;j<size;j++)

{

if(array[i]==array[j])

{ count++;

for(k=j;k<size;k++)

array[k]=array[k+1];

}

}

}

size=size-count;

for (i = 0; i < size; i++) {

printf("%d\n", array[i]);

}

}