**A13 - WAP to find 3rd largest element in an array**

/\*

Name : Nestin Gregorios Sunny

Date : 12.05.2025

Description :

Find third largest element in the array

Sample Input :

Enter the size of the array : 5

Enter the elements into the array: 5 1 4 2 8

Sample Output :

Third largest element of the array is 4

\*/

#include <stdio.h>

//function declaration

void read(int [], int);

int third\_largest(int [], int);

int main()

{

int size, ret;

//Read size from the user

printf("Enter the size of the array :");

scanf("%d", &size);

int arr[size];

//Read elements into the array

printf("Enter the elements into the array: ");

read(arr, size); //read elements to array

//funtion call

ret = third\_largest(arr, size);

printf("Third largest element of the array is %d\n", ret);

}

//function definition

void read(int arr[], int size)

{

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

{

scanf("%d",&arr[i]);

}

}

int third\_largest(int \*arr, int size)

{

int i, largest, sec\_largest, third\_largest;

//assuming for comparing

largest = arr[0];

sec\_largest = arr[1];

third\_largest = arr[2];

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

{

if(arr[i] > largest) //condition for 1st largest

{

third\_largest = sec\_largest;

sec\_largest = largest;

largest = arr[i];

}

else if(arr[i] > sec\_largest && arr[i] != largest) //condition for 2nd largest

{

third\_largest = sec\_largest;

sec\_largest = arr[i];

}

else if(arr[i] > third\_largest && arr[i] != sec\_largest && arr[i] != largest) //condition for third largest

{

third\_largest = arr[i];

}

}

return third\_largest;

}

**A14 - WAP to find 2nd largest element in an array**

/\*

Name : Nestin Gregorios Sunny

Date : 12.05.2025

Description :

Find 2nd largest element from the array

Sample Input :

Enter the size of the Array : 5

Enter the elements into the array: 5 1 4 2 8

Sample Output :

Second largest element of the array is 5

\*/

#include <stdio.h>

int sec\_largest(int [], int);

int main()

{

int i, size, ret;

//Read size from the user

//printf("Enter the size of the array :");

scanf("%d", &size);

int arr[size];

//Read elements into the array

//printf("Enter the elements of the array: ");

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

{

scanf("%d",&arr[i]);

}

//funtion call

ret = sec\_largest(arr, size);

printf("Second largest element of the array is %d\n", ret);

}

int sec\_largest(int \*arr, int size)

{

int i, largest, sec\_largest, smallest;

largest = arr[0];

smallest = arr[0];

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

{

if(arr[i] > largest) //find largest

{

largest = arr[i];

}

if(arr[i] < smallest) //find smallest

{

smallest = arr[i];

}

}

sec\_largest = smallest;

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

{

if(largest > arr[i] && arr[i] > sec\_largest) //find sec\_largest

{

sec\_largest = arr[i];

}

}

return sec\_largest;

}

**A15 - WAP to remove duplicate elements in a given array**

/\*

Name : Nestin Gregorios Sunny

Date : 10.05.2025

Description :

Remove duplicates from user read array

Sample Input :

Enter the size: 5

Enter the elements into the array: 5 1 3 1 5

Sample Output :

After removing duplicates: 5 1 3

\*/

#include <stdio.h>

//function declaration

void fun(int arr1[], int size, int arr2[], int \*new\_size);

void read(int arr[], int size);

int main()

{

int i,size;

printf("Enter the size: ");

scanf("%d",&size);

int arr1[size];

printf("Enter elements into the array: ");

read(arr1,size); //calling read() to read array elements

int arr2[size];

int count = 1; //to count how many unique values are there

fun(arr1, size, arr2, &count); //calling fun() to remove duplicates and print unique array in arr2

return 0;

}

//function definition

void read(int arr[], int size)

{

int i;

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

{

scanf("%d",&arr[i]);

}

}

void fun(int arr1[], int size, int arr2[], int \*new\_size)

{

int i, j;

arr2[0] = arr1[0]; //1st element of both arrays will be unique

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

{

int flag = 0;

for(j = 0; j < \*new\_size ; j++)

{

if(arr1[i] == arr2[j]) //condition to detect duplicate element

{

flag = 1;

break;

}

}

if(flag == 0) //if not duplicate assign that element to arr2[]

{

arr2[\*new\_size] = arr1[i];

\*new\_size = \*new\_size + 1;

}

}

printf("After removing duplicates: ");

for(i = 0; i< \*new\_size; i++)

{

printf("%d ",arr2[i]);

}

}

**A16 - Print the values in sorted order without modifying or copying array**

/\*

Name : Nestin Gregorios Sunny

Date : 14.05.2025

Description :

Print the values in sorted order without modifying or copying the array

Sample Input :

Enter the size : 5

Enter the 5 elements

10 1 3 8 -1

Sample Output :

After sorting: -1 1 3 8 10

Orginal

\*/

#include <stdio.h>

//function declaration

void print\_sort(int [], int);

int main()

{

int size, iter;

//printf("Enter the size of the array : ");

scanf("%d", &size);

int arr[size];

//printf("Enter the %d elements\n",size);

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

{

scanf("%d", &arr[iter]);

}

printf("After sorting: ");

print\_sort(arr, size); //function call

printf("\nOriginal array values ");

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

{

printf("%d ",arr[iter]);

}

}

//function definition

void print\_sort(int \*arr, int size)

{

int i, j, large, small, sec\_small;

large = arr[0];

small =arr[0];

for(i = 0; i < size; i++) //find largest and smallest

{

if(arr[i] > large)

{

large = arr[i];

}

if(arr[i] < small)

{

small = arr[i];

}

}

sec\_small = large; // assign sec\_small to largest

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

{

printf("%d ",small);

sec\_small = large; //after every iteration sec\_small get updated to another value

//to avoid that we are again assigning Sec\_small to largest

for(j = 0; j < size; j++) //loop to find second smallest

{

if(arr[j] > small && arr[j] < sec\_small)

{

sec\_small = arr[j];

}

}

small = sec\_small; //small is updated to sec\_small to print next small values as sec\_small

}

}

**A17 - WAP to check whether a given number is prime or not using function.**

/\*

Name : Nestin Gregorios Sunny

Date : 14.05.2025

Description :

Read a number from user and check whether its prime or not.

Sample Input :

Enter a number: 2

Sample Output :

2 is a prime number

\*/

#include <stdio.h>

//function declaration

int is\_prime(int);

int main()

{

int num;

//printf("Enter a number: ");

scanf("%d",&num);

int prime = is\_prime(num); //function call

if(num < 0)

{

printf("Invalid input");

}

else

{

if(prime)

{

printf("%d is a prime number",num);

}

else

{

printf("%d is not a prime number",num);

}

}

return 0;

}

//function definition

int is\_prime(int n)

{

int i,flag = 1;

if(n <= 1)

{

flag = 0;

}

else

{

for(i = 2; i < n; i++)

{

if(n % i ==0)

{

flag = 0;

break;

}

}

}

return flag;

}