



C Online Compiler

[Interactive C Course](#)

main.c

Run

Output

Clear

```
1 #include <stdio.h>
2 int main() {
3     int n;
4     double arr[100];
5     printf("Enter the number of elements (1 to 100): ");
6     scanf("%d", &n);
7
8     for (int i = 0; i < n; ++i) {
9         printf("Enter number%d: ", i + 1);
10        scanf("%lf", &arr[i]);
11    }
12
13    // storing the largest number to arr[0]
14    for (int i = 1; i < n; ++i) {
15        if (arr[0] < arr[i]) {
16            arr[0] = arr[i];
17        }
18    }
19
20    printf("Largest element = %.2lf", arr[0]);
21
22    return 0;
23 }
```

```
/tmp/D8svL9Q6Hs.o
Enter the number of elements (1 to 100): 5
Enter number1: 0
Enter number2: 7
Enter number3: 4
Enter number4: 09
Enter number5: 089
Largest element = 89.00
```

main.c



Run

Output

Clear

```
1 #include <stdio.h>
2
3 void findSecondLargestAndSmallest(int arr[], int size) {
4     int firstLargest = arr[0], secondLargest = arr[0];
5     int firstSmallest = arr[0], secondSmallest = arr[0];
6
7     for (int i = 1; i < size; i++) {
8         if (arr[i] > firstLargest) {
9             secondLargest = firstLargest;
10            firstLargest = arr[i];
11        }
12        else if (arr[i] > secondLargest && arr[i] != firstLargest) {
13            secondLargest = arr[i];
14        }
15
16        if (arr[i] < firstSmallest) {
17            secondSmallest = firstSmallest;
18            firstSmallest = arr[i];
19        }
20        else if (arr[i] < secondSmallest && arr[i] != firstSmallest) {
21            secondSmallest = arr[i];
22        }
23    }
```

```
/tmp/D8svL9Q6Hs.o
Second largest element is 8
Second smallest element is 2
```

main.c

Run

Output

Clear

```
1 #include <stdio.h>
2
3 int main() {
4     int n, i, j, odd_element;
5     printf("Enter the size of the array: ");
6     scanf("%d", &n);
7     int arr[n];
8     printf("Enter the array elements (containing only two different
    elements): ");
9     for(i=0; i<n; i++) {
10         scanf("%d", &arr[i]);
11     }
12     for(i=0; i<n; i++) {
13         int count = 0;
14         for(j=0; j<n; j++) {
15             if(arr[i] == arr[j]) {
16                 count++;
17             }
18         }
19         if(count % 2 != 0) {
20             odd_element = arr[i];
21             break;
22     }
```

```
/tmp/D8svL9Q6Hs.o
Enter the size of the array: 19
Enter the array elements (containing only two different elements): 23
45
11
123
123
245
657
79
875
The odd element is: 2355
```

Activate Windows
Go to Settings to activate Windows.

main.c



Run

Output

Clear

```
1  #include <stdio.h>
2  #define MAX_SIZE 100
3
4  int main()
5  {
6      int arr[MAX_SIZE];
7      int i, size, pos;
8
9      /* Input size and element in array */
10     printf("Enter size of the array : ");
11     scanf("%d", &size);
12     printf("Enter elements in array : ");
13     for(i=0; i<size; i++)
14     {
15         scanf("%d", &arr[i]);
16     }
17
18     /* Input element position to delete */
19     printf("Enter the element position to delete : ");
20     scanf("%d", &pos);
21
22
23     /* Invalid delete position */
```

```
/tmp/D8svL9Q6Hs.o
Enter size of the array : 5
Enter elements in array : 10 20 30 40 50
Enter the element position to delete : 3
Elements of array after delete are : 10 20 40 50 |
```

Activate Windows
Go to Settings to activate Windows.

main.c

Run

Output

Clear

```
1 #include <stdio.h>
2
3 int duplicate_element ( int arr[], int num)
4 {
5     // check num is equal to 0 and num == 1
6     if (num == 0 || num == 1)
7         return num;
8
9     // create temp array to store same number
10    int temp [num];
11
12    // declare variable
13    int i, j = 0;
14
15    // use for loop to check duplicate element
16    for (i = 0; i < num - 1; i++)
17    {
18        // check the element of i is not equal to (i + 1) next
19        // element
20        if (arr [i] != arr[i + 1])
21            temp[j++] = arr[i];
22    }
23    temp[i++] = arr[num - 1];
```

```
/tmp/tpLaTG382Q.o
Define the no. of elements of the array: 5
Enter the elements: 3
5435
2345
78965
2351
Elements before removing duplicates: 3 5435 2345 78965 2351
Display array's elements after removing duplicates: 3 5435 2345 78965
23513
```

Activate Windows
Go to Settings to activate Windows.

main.c



Run

Output

Clear

```
9  int n;
10 // Inputting the size of the array
11 printf("Enter the size of the array: ");
12 scanf("%d", &n);
13
14 // Inputting the array
15 printf("Enter an array: ");
16 for (int i = 0; i < n; i++){
17     scanf("%d", &arr[i]);
18 }
19
20 // Printing the reverse of the array
21 printf("Reversed array: ");
22 for (int i = n-1; i >= 0; i--){
23     printf("%d ", arr[i]);
24 }
25
26 return 0;
27 }
28
29 // reverse array in c
30 // reverse an array in c
31
```

```
/tmp/tpLaTG382Q.o
Enter the size of the array: 10
Enter an array: 11 12 14 15 12 13 13 18 14 18
Reversed array: 18 14 18 13 13 12 15 14 12 11 |
```

Activate Windows
Go to Settings to activate Windows.

main.c



Run

Output

Clear

```
37         arr3[in] = arr2[j];
38         j++;
39     }
40 }
41 else if (i < n)
42 {
43     arr3[in] = arr1[i];
44     i++;
45 }
46 else
47 {
48     arr3[in] = arr2[j];
49     j++;
50 }
51 }
52 printf("The merged array is: \n");
53 for (in = 0; in < n + k; in++)
54 {
55     printf("%d ", arr3[in]);
56 }
57 printf("\n");
58 return 0;
59 }
```

```
/tmp/tpLaTG382Q.o
Enter the size of the first array: 4
Enter the elements of the first array:
12
12
56
78
Enter the size of the second array: 3
Enter the elements of the second array:
12
657
90
The merged array is:
12 12 12 56 78 657 90
```

Activate Windows
Go to Settings to activate Windows.

main.c



Run

Output

Clear

```
1  /- for (i = 2; i <= n / 2; i++) {
2      if (n % i == 0) {
3          return 1; // Number is composite
4      }
5  }
6  return 0; // Number is not composite
7  }
8
9  int main() {
10     int arr[100], n, i, count = 0;
11     printf("Enter the size of array: ");
12     scanf("%d", &n);
13     printf("Enter the array elements:\n");
14     for (i = 0; i < n; i++) {
15         scanf("%d", &arr[i]);
16         if (isComposite(arr[i])) {
17             count++; // Increment count if element is composite
18         }
19     }
20     printf("Number of composite numbers in the array: %d", count);
21     return 0;
22 }
```

```
/tmp/tpLaTG382Q.o
Enter the size of array: 4
Enter the array elements:
12 45 78 34
Number of composite numbers in the array: 4
```

Activate Windows
Go to Settings to activate Windows.