



**Submitted by –**

**Name: Sabbir Ahmed Shisir**

**Reg: 23201163**

**Sec : C-2**

**Course Code : CSE-108.**

**Submitted to-**

**Zaima Sartaj Taheri  
ma'am**

**Lecturer in UAP.**

//(1)- Find the the maximum & min of an array.

```
#include <stdio.h>
```

```
int main() {
```

```
    int n;
```

```
    printf("Enter the num of elements: ");
```

```
    scanf("%d", &n);
```

```
    int arr[n];
```

```
    printf("Enter the elements of the array:\n");
```

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

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

```
    }
```

```
    int max = arr[0];
```

```
    int min = arr[0];
```

```
    for (int i = 1; i < n; i++) {
```

```
        if (arr[i] > max) {
```

```
            max = arr[i];
```

```
        }
```

```
        if (arr[i] < min) {
```

```
            min = arr[i];
```

```
        }
```

```
    }
```

```
    printf("Maximum element in the array is: %d\n", max);
```

```
    printf("Minimum element in the array is: %d\n", min);
```

```
    return 0;
```

```
}
```



main.c



Share

Run

Output

Clear

```
1 //1)- Find tyhe the maximum & min of an array.
2 #include <stdio.h>
3 int main() {
4     int n;
5
6     printf("Enter the num of elements: ");
7     scanf("%d", &n);
8
9     int arr[n];
10    printf("Enter the elements of the array:\n");
11    for (int i = 0; i < n; i++) {
12        scanf("%d", &arr[i]);
13    }
14
15    int max = arr[0];
16    int min = arr[0];
17
18    for (int i = 1; i < n; i++) {
19        if (arr[i] > max) {
20            max = arr[i];
21        }
22        if (arr[i] < min) {
23            min = arr[i];
24        }
25    }
26    printf("Maximum element in the array is: %d\n", max);
27    printf("Minimum element in the array is: %d\n", min);
28    return 0; }
```

```
/tmp/ZTB6EtiExp.o
Enter the num of elements: 5
Enter the elements of the array:
45
56
5
20
55
Maximum element in the array is: 56
Minimum element in the array is: 5

=== Code Execution Successful ===
```

// (2)-Array Reverse.

```
#include <stdio.h>
```

```
int main() {
```

```
    int n;
```

```
    printf("Enter the num of elements : ");
```

```
    scanf("%d", &n);
```

```
    int arr[n];
```

```
    printf("Enter the elements of the array:\n");
```

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

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

```
    }
```

```
    for (int i = 0; i < n / 2; i++) {
```

```
        int temp = arr[i];
```

```
        arr[i] = arr[n - i - 1];
```

```
        arr[n - i - 1] = temp;
```

```
    }
```

```
    printf("Reversed array:\n");
```

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

```
        printf("%d ", arr[i]);
```

```
    }
```

```
    return 0;
```

```
}
```



**Struggling with high humidity levels? ...**

Say goodbye to sticky and uncomfortable air in your...

Sponsored by: Air Conditioning Italy

[Learn more](#)

[Programiz PRO >](#)

main.c	Run	Output	Clear
<pre>1 // (2)-Array Reverse. 2 #include &lt;stdio.h&gt; 3 4 int main() { 5     int n; 6 7     printf("Enter the num of elements : "); 8     scanf("%d", &amp;n); 9 10    int arr[n]; 11    printf("Enter the elements of the array:\n"); 12    for (int i = 0; i &lt; n; i++) { 13        scanf("%d", &amp;arr[i]); 14    } 15 16    for (int i = 0; i &lt; n / 2; i++) { 17        int temp = arr[i]; 18        arr[i] = arr[n - i - 1]; 19        arr[n - i - 1] = temp; 20    } 21 22    printf("Reversed array:\n"); 23    for (int i = 0; i &lt; n; i++) { 24        printf("%d ", arr[i]); 25    } 26    return 0; 27 }</pre>		<pre>/tmp/DTqQGHseK6.o Enter the number of elements in the array: 5 Enter the elements of the array: 12 5 6 23 65 Reversed array: 65 23 6 5 12  === Code Execution Successful ===</pre>	

// (3)-Program to cyclically rotate an array by one.

```
#include <stdio.h>
```

```
int main() {
```

```
    int arr[] = {1, 2, 3, 4, 5};
```

```
    int n = sizeof(arr) / sizeof(arr[0]);
```

```
    printf("Original array: ");
```

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

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

```
    }
```

```
    int last = arr[n - 1];
```

```
    for (int i = n - 1; i > 0; i--) {
```

```
        arr[i] = arr[i - 1];
```

```
    }
```

```
    arr[0] = last;
```

```
    printf("Array after rotation:");
```

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

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

```
    }
```

```
    // printf("\n");
```

```
    return 0;
```

```
}
```



C Online Compiler

Premium Coding  
Courses by Programiz



Programiz PRO >

main.c



Run

Output

Clear

```
1 // (3)-Program to cyclically rotate an array by one.
```

```
2 #include <stdio.h>
```

```
3 int main() {
```

```
4     int arr[] = {1, 2, 3, 4, 5};
```

```
5     int n = sizeof(arr) / sizeof(arr[0]);
```

```
6
```

```
7     printf("Original array: ");
```

```
8     for (int i = 0; i < n; i++) {
```

```
9         printf("%d \n", arr[i]);
```

```
10     }
```

```
11
```

```
12     int last = arr[n - 1];
```

```
13     for (int i = n - 1; i > 0; i--) {
```

```
14         arr[i] = arr[i - 1];
```

```
15     }
```

```
16
```

```
17     arr[0] = last;
```

```
18
```

```
19     printf("Array after rotation:");
```

```
20     for (int i = 0; i < n; i++) {
```

```
21         printf("%d \n", arr[i]);
```

```
22     }
```

```
23     // printf("\n");
```

```
24
```

```
25     return 0;
```

```
26 }
```

/tmp/UGlcQp0X60.o

Original array: 1

2

3

4

5

Array after rotation:5

1

2

3

4

=== Code Execution Successful ===

//4-Short an array elements (in Ascending order)....

```
#include <stdio.h>
```

```
int main() {
```

```
    int temp ,n ;
```

```
    int arr[] = {5, 2, 9, 1, 5, 6};
```

```
    n = sizeof(arr) / sizeof(arr[0]);
```

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

```
        for (int j = i + 1; j < n; j++) {
```

```
            if (arr[i] > arr[j]) { // Swap the elements
```

```
                temp = arr[i];
```

```
                arr[i] = arr[j];
```

```
                arr[j] = temp;
```

```
            }
```

```
        }
```

```
    }
```

```
    printf("Sorted array in Ascending order: \n");
```

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

```
    {
```

```
        printf("%d ", arr[i]);
```

```
    }
```

```
    return 0;
```

```
}
```





C Online Compiler

Premium Coding  
Courses by Programiz



Programiz PRO >

main.c



Run

Output

Clear

```
1 //4-Short an array elements (in Ascending order)...\n2 #include <stdio.h>\n3 int main() {\n4\n5     int temp,n ;\n6     int arr[] = {5, 2, 9, 1, 5, 6};\n7     n = sizeof(arr) / sizeof(arr[0]);\n8\n9     for (int i = 0; i < n; i++) {\n10         for (int j = i + 1; j < n; j++) {\n11             if (arr[i] > arr[j]) { // Swap the elements\n12                 temp = arr[i];\n13                 arr[i] = arr[j];\n14                 arr[j] = temp;\n15             }\n16         }\n17     }\n18\n19     printf("Sorted array in Ascending order: \\n");\n20     for (int i = 0; i < n; i++)\n21     {\n22         printf("%d ", arr[i]);\n23     }\n24\n25     return 0;\n26 }\n27
```

```
/tmp/gFYRU2zfcs.o\nSorted array in Ascending order:\n1 2 5 5 6 9\n\n=== Code Execution Successful ===
```



11:46 PM

9/28/2024



//(5)-Find duplicate elements in an array.

```
#include <stdio.h>
```

```
int main() {
```

```
    int i, j;
```

```
    int arr[] = {1, 2, 3, 4, 2, 5, 1, 6};
```

```
    int n= sizeof(arr)/ sizeof(arr[0]);
```

```
    printf("Duplicate elements are: ");
```

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

```
        for (j = i + 1; j < n; j++) {
```

```
            if (arr[i] == arr[j]) {
```

```
                printf("%d ", arr[i]);
```

```
            }
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```



main.c



Share

Run

Output

Clear

```
1 //5)-Find duplicate elements in an array.
2 #include <stdio.h>
3 int main() {
4     int i, j;
5     int arr[] = {1, 2, 3, 4, 2, 5, 1, 6};
6     int n= sizeof(arr)/ sizeof(arr[0]);
7
8     printf("Duplicate elements are: ");
9     for (i = 0; i < n; i++) {
10         for (j = i + 1; j < n; j++) {
11             if (arr[i] == arr[j]) {
12                 printf("%d ", arr[i]);
13             }
14         }
15     }
16     return 0;
17 }
18
19
```

/tmp/fPSZDG7Zwb.o

Duplicate elements are: 1 2

=== Code Execution Successful ===



//(6)-Count the occurrence of each element of in a array.

```
#include <stdio.h>
```

```
int main() {
```

```
int A[]={1,2,3,1,1,4};
```

```
int a1=0, a2=0, a3=0, a4=0;
```

```
for(int i=0; i<=5; i++){
```

```
if(A[i]==1){
```

```
a1++;
```

```
}
```

```
else if(A[i]==2){
```

```
a2++;
```

```
}
```

```
else if(A[i]==3){
```

```
a3++;
```

```
}
```

```
else if(A[i]==4){
```

```
a4++;
```

```
}
```

```
}
```

```
printf("occurrence of 1- %d \n",a1);
```

```
printf("occurrence of 2- %d\n" ,a2);
```

```
printf("occurrence of 3- %d\n" ,a3);
```

```
printf("occurrence of 4- %d\n" ,a4);
```

```
return 0;
```

```
}
```

main.c

Share Run

Output Clear

```
1 //6)-Count the occurrence of each element of in a array.
2 #include <stdio.h>
3 int main() {
4     int A[]={1,2,3,1,1,4};
5     int a1=0, a2=0, a3=0, a4=0;
6
7     for(int i=0; i<5; i++){
8         if(A[i]==1){
9             a1++;
10        }
11        else if(A[i]==2){
12            a2++;
13        }
14        else if(A[i]==3){
15            a3++;
16        }
17        else if(A[i]==4){
18            a4++;
19        }
20
21    }
22    printf("occurrence of 1- %d \n",a1);
23    printf("occurrence of 2- %d\n", a2);
24    printf("occurrence of 3- %d\n", a3);
25    printf("occurrence of 4- %d\n", a4);
26
27    return 0;
28 }
29
```

/tmp/RBcDoYd5uV.o

occurrence of 1- 3

occurrence of 2- 1

occurrence of 3- 1

occurrence of 4- 1

=== Code Execution Successful ===

//(7)- Sort the array 0s, 1s, 2s.

```
#include <stdio.h>
```

```
int main() {
```

```
    int temp ,n ;
```

```
    int arr[] = {2, 0, 1, 2, 0, 1};
```

```
    n = sizeof(arr) / sizeof(arr[0]);
```

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

```
        for (int j = i + 1; j < n; j++) {
```

```
            if (arr[i] > arr[j]) { // Swap the elements
```

```
                temp = arr[i];
```

```
                arr[i] = arr[j];
```

```
                arr[j] = temp;
```

```
            }
```

```
        }
```

```
    }
```

```
    printf("Sort the array:");
```

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

```
    {
```

```
        printf("%d ", arr[i]);
```

```
    }
```

```
    return 0;
```

```
}
```



C Online Compiler

Programiz PRO >

main.c



Output

Clear

```
1 //7)- Sort the array 0s, 1s,2s.
2 #include <stdio.h>
3 int main() {
4
5     int temp ,n ;
6     int arr[] = {2, 0, 1, 2, 0, 1};
7     n = sizeof(arr) / sizeof(arr[0]);
8
9     for (int i = 0; i < n ; i++) {
10         for (int j = i + 1; j < n; j++) {
11             if (arr[i] > arr[j]) { // Swap the elements
12                 temp = arr[i];
13                 arr[i] = arr[j];
14                 arr[j] = temp;
15             }
16         }
17     }
18
19     printf("Sort the array:");
20     for (int i = 0; i < n; i++)
21     {
22         printf("%d ", arr[i]);
23     }
24     return 0;
25 }
26
```

/tmp/I2UUkx7eMF.o

Sort the array:0 0 1 1 2 2

=== Code Execution Successful ===

//(8)-Move all negative numbers to beginning and positive to end with constant extra space.

```
#include <stdio.h>
```

```
int main() {  
    int arr[] = {-2, 5, -3, 9, 10, -6, -7};  
    int n = 7;  
    int temp;  
  
    for (int i = 0; i < n; i++) {  
        for (int j = i + 1; j < n; j++) {  
            if (arr[i] > 0 && arr[j] < 0) {  
  
                temp = arr[i];  
                arr[i] = arr[j];  
                arr[j] = temp;  
            }  
        }  
    }  
    printf("Rearranged array: \n");  
    for (int i = 0; i < n; i++) {  
        printf("%d ", arr[i]);  
    }  
    printf("\n");  
    return 0;  
}
```



main.c



Run

Output

1 // (8)-Move all negative numbers to beginning and positive to end with constant extra space.

2 #include <stdio.h>

3 int main() {

4 int arr[] = {-2, 5, -3, 9, 10, -6, -7};

5 int n = 7;

6 int temp;

7

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

9 for (int j = i + 1; j < n; j++) {

10 if (arr[i] > 0 && arr[j] < 0) {

11

12 temp = arr[i];

13 arr[i] = arr[j];

14 arr[j] = temp;

15 }

16 }

17 }

18 printf("Rearranged array: \n");

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

20 printf("%d ", arr[i]);

21 }

22 printf("\n");

23 return 0;

24 }

25

26

27

/tmp/QRnY11yUc.o

Rearranged array:

-2 -3 -6 -7 10 5 9

=== Code Execution Successful ===

//(9)- Find the row with maximum number of 1s

```
#include <stdio.h>
```

```
int main() {
```

```
    int a[4][4] = {
```

```
        {0, 0, 0, 1},
```

```
        {1, 0, 1, 1},
```

```
        {0, 1, 1, 1},
```

```
        {1, 1, 1, 1}
```

```
    };
```

```
    int c1 = 0, c2 = 0, c3 = 0, c4 = 0;
```

```
    for (int j = 0; j < 4; j++) {
```

```
        if (a[0][j] == 1) {
```

```
            c1++;
```

```
        }
```

```
        if (a[1][j] == 1) {
```

```
            c2++;
```

```
        }
```

```
        if (a[2][j] == 1) {
```

```
            c3++;
```

```
        }
```

```
        if (a[3][j] == 1) {
```

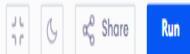
```
            c4++;
```

```
        }
```

```
    }
```

```
if (c1 > c2 && c1 > c3 && c1 > c4) {  
    printf("The maximum num of 1's is in row: %d times\n", c1);  
}  
else if (c2 > c1 && c2 > c3 && c2 > c4) {  
    printf("The maximum num of 1's is in row: %d times\n", c2);  
}  
else if (c3 > c1 && c3 > c2 && c3 > c4) {  
    printf("The maximum num of 1's is in row: %d times\n", c3);  
}  
else if (c4 > c1 && c4 > c2 && c4 > c3) {  
    printf("The maximum num of 1's is in row: %d times\n", c4);  
}  
else {  
    printf("There are multiple rows with the same maximum number of 1s.\n");  
}  
return 0;  
}
```

main.c



Output

```
1 //9)- Find the row with maximum number of 1s
2 #include <stdio.h>
3 int main() {
4     int a[4][4] = {
5         {0, 0, 0, 1},
6         {1, 0, 1, 1},
7         {0, 1, 1, 1},
8         {1, 1, 1, 1}
9     };
10    int c1 = 0, c2 = 0, c3 = 0, c4 = 0;
11
12    for (int j = 0; j < 4; j++) {
13        if (a[0][j] == 1) {
14            c1++;
15        }
16        if (a[1][j] == 1) {
17            c2++;
18        }
19        if (a[2][j] == 1) {
20            c3++;
21        }
22        if (a[3][j] == 1) {
23            c4++;
24        }
25    }
26
27    if (c1 > c2 && c1 > c3 && c1 > c4) {
28        printf("The maximum num of 1's is in row: %d times\n", c1);
29    }
30    else if (c2 > c1 && c2 > c3 && c2 > c4) {
31        printf("The maximum num of 1's is in row: %d times\n", c2);
32    }
33    else if (c3 > c1 && c3 > c2 && c3 > c4) {
34        printf("The maximum num of 1's is in row: %d times\n", c3);
35    }
36    else if (c4 > c1 && c4 > c2 && c4 > c3) {
37        printf("The maximum num of 1's is in row: %d times\n", c4);
38    }
39    else {
40        printf("There are multiple rows with the same maximum number of 1s.\n");
41    }
42    return 0;
43 }
```

/tmp/IPRKi6j5Zc.o  
The maximum num of 1's is in row: 4 times  
  
=== Code Execution Successful ===

//(10)-Majority Element- Given an array arr. Find the majority element in the array. If no majority exists, return -1. A majority element in an array is an element that appears strictly more than  $\text{arr.size() / 2}$  times in the array.

```
#include <stdio.h>
```

```
int main() {
```

```
    int A[] = {3, 3, 4, 2, 4, 4, 2, 4};
```

```
    int n= sizeof(A)/ sizeof(A[0]);
```

```
    int c1 = 0, c2 = 0, c3 = 0;
```

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

```
        if (A[i] == 1) {
```

```
            c1++;
```

```
        }
```

```
        else if (A[i] == 2) {
```

```
            c2++;
```

```
        }
```

```
        else if (A[i] == 3) {
```

```
            c3++;
```

```
        }
```

```
    }
```

```
    if (c1 > 3) {
```

```
        printf("Maximum count is of 1: %d\n", c1);
```

```
    }
```

```
    else if (c2 > 3) {
```

```
        printf("Maximum count is of 2: %d\n", c2);
```

```
    }
```

```
    else if (c3 > 3) {
```

```
        printf("Maximum count is of 3: %d\n", c3);
```

Classwork for CSE x Top 50 Array Coding x Majority Element x Arithmetic, Geometry x Online C Compiler x Lab Report-3 x + -

→ ↻ 🏠 programiz.com/c-programming/online-compiler/ 🔍 ☆ 🎵 S ⋮

programiz C Online Compiler



//(11)- Sort an array in wave form.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int array[] = {7, 9, 2, 1, 6, 17};
```

```
    int temp;
```

```
    int i, j;
```

```
    for (i = 0; i < 6 - 1; i++)
```

```
    {
```

```
        for (j = i + 1; j < 6; j++)
```

```
        {
```

```
            if (array[i] > array[j])
```

```
            {
```

```
                temp = array[i];
```

```
                array[i] = array[j];
```

```
                array[j] = temp;
```

```
            }
```

```
        }
```

```
    }
```

```
    for (i = 0; i < 6; i += 2)
```

```
    {
```

```
        temp = array[i];
```

```
        array[i] = array[i + 1];
```

```
        array[i + 1] = temp;
```

```
}
```

```
for (i = 0; i < 6; i++)
```

```
{
```





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

```
}
```

```
return 0;
```

```
}
```



main.c	   	Output
<pre> 1 // (11)- Sort an array in wave form. 2 #include &lt;stdio.h&gt; 3 int main() 4 { 5     int array[] = {7, 9, 2, 1, 6, 17}; 6     int temp; 7     int i, j; 8 9     for (i = 0; i &lt; 6 - 1; i++) 10    { 11        for (j = i + 1; j &lt; 6; j++) 12        { 13            if (array[i] &gt; array[j]) 14            { 15                temp = array[i]; 16                array[i] = array[j]; 17                array[j] = temp; 18            } 19        } 20    } 21 22    for (i = 0; i &lt; 6; i += 2) 23    { 24        temp = array[i]; 25        array[i] = array[i + 1]; 26        array[i + 1] = temp; 27    } 28 29    for (i = 0; i &lt; 6; i++) 30    { 31        printf("%d ", array[i]); 32    } 33    return 0; 34 } </pre>	<pre> /tmp/0wseNExcJZ.o 2 1 7 6 17 9  === Code Execution Successful === </pre>	

