

1. Reverse an Array

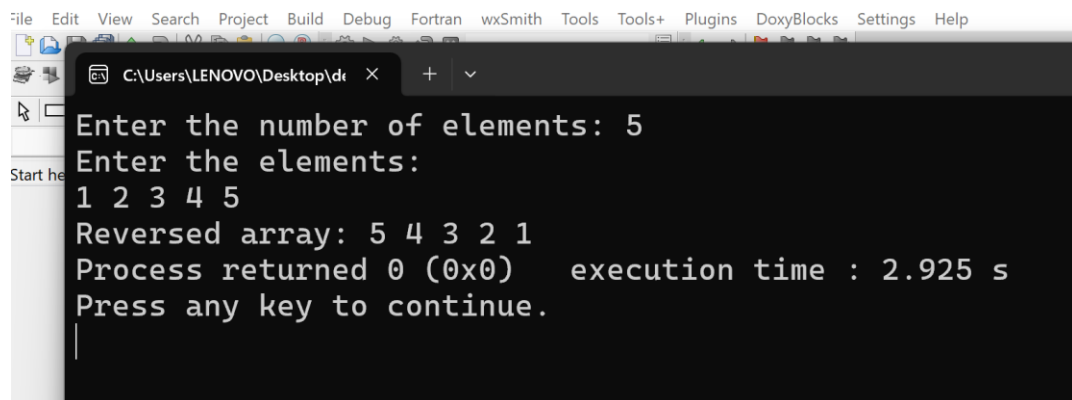
Description:

- Print the array elements in reverse order.

Code:

```
1  #include <stdio.h>
2
3  int main() {
4      int n;
5      printf("Enter the number of elements: ");
6      scanf("%d", &n);
7      int arr[n];
8      printf("Enter the elements:\n");
9      for (int i = 0; i < n; i++) {
10         scanf("%d", &arr[i]);
11     }
12     printf("Reversed array: ");
13     for (int i = n - 1; i >= 0; i--) {
14         printf("%d ", arr[i]);
15     }
16     return 0;
17 }
18
```

Example Output:



The screenshot shows a Windows command prompt window with the following text:

```
Enter the number of elements: 5
Enter the elements:
1 2 3 4 5
Reversed array: 5 4 3 2 1
Process returned 0 (0x0)   execution time : 2.925 s
Press any key to continue.
```

The background shows a code editor window with the file path `C:\Users\LENOVO\Desktop\de`.

2. Sum and Average of Array

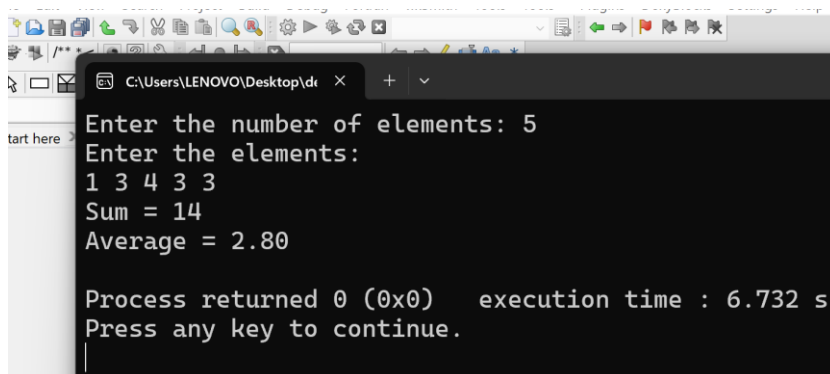
Description:

- Calculate and print the sum and average of array elements.

Code:

```
1  #include <stdio.h>
2
3  int main() {
4      int n, sum = 0;
5      printf("Enter the number of elements: ");
6      scanf("%d", &n);
7
8      int arr[n];
9      printf("Enter the elements:\n");
10     for (int i = 0; i < n; i++) {
11         scanf("%d", &arr[i]);
12         sum += arr[i];
13     }
14
15     float average = (float)sum / n;
16     printf("Sum = %d\n", sum);
17     printf("Average = %.2f\n", average);
18     return 0;
19 }
20
```

Example Output:



The screenshot shows a Windows command prompt window with the following text:

```
Enter the number of elements: 5
Enter the elements:
1 3 4 3 3
Sum = 14
Average = 2.80

Process returned 0 (0x0)   execution time : 6.732 s
Press any key to continue.
```

3. Count Duplicates in Array

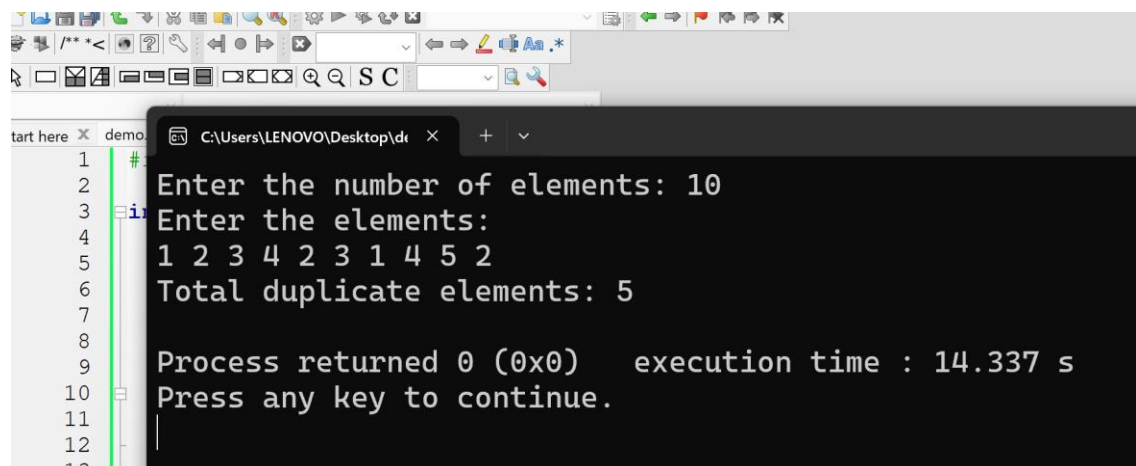
Description:

- Count the number of duplicate elements in the array.

Code:

```
1 #include <stdio.h>
2
3 int main() {
4     int n, count = 0;
5     printf("Enter the number of elements: ");
6     scanf("%d", &n);
7
8     int arr[n];
9     printf("Enter the elements:\n");
10    for (int i = 0; i < n; i++) {
11        scanf("%d", &arr[i]);
12    }
13
14    for (int i = 0; i < n; i++) {
15        for (int j = i + 1; j < n; j++) {
16            if (arr[i] == arr[j]) {
17                count++;
18                break;
19            }
20        }
21    }
22
23    printf("Total duplicate elements: %d\n", count);
24    return 0;
25 }
26
```

Example Output:



The screenshot shows a Windows command prompt window with a dark background. The user has entered '10' for the number of elements and then entered the array elements: '1 2 3 4 2 3 1 4 5 2'. The program outputs 'Total duplicate elements: 5'. Below this, it shows 'Process returned 0 (0x0) execution time : 14.337 s' and 'Press any key to continue.' The command prompt window is titled 'C:\Users\LENOVO\Desktop\di' and has a tab labeled 'demo'.

```
1 #
2
3 Enter the number of elements: 10
4 Enter the elements:
5 1 2 3 4 2 3 1 4 5 2
6 Total duplicate elements: 5
7
8
9 Process returned 0 (0x0) execution time : 14.337 s
10 Press any key to continue.
11
12
13
```

4. Count Frequency of Each Element

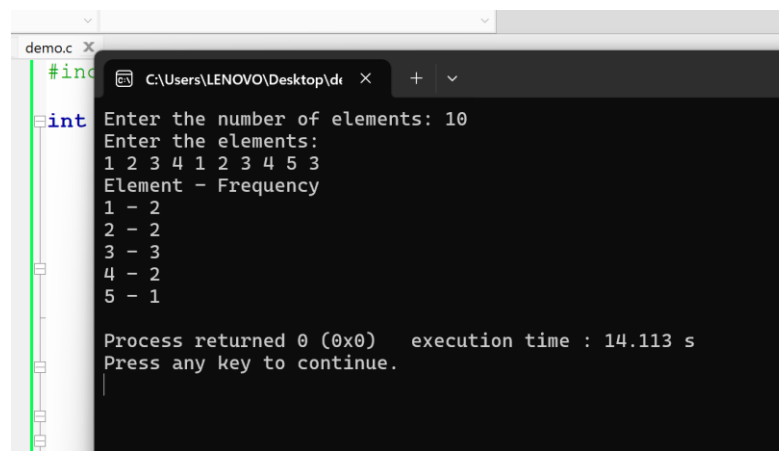
Description:

Count the frequency of each element in the array.

Code:

```
1  #include <stdio.h>
2
3  int main() {
4      int n;
5      printf("Enter the number of elements: ");
6      scanf("%d", &n);
7      int arr[n];
8      printf("Enter the elements:\n");
9      for (int i = 0; i < n; i++) {
10         scanf("%d", &arr[i]);
11     }
12     printf("Element - Frequency\n");
13     for (int i = 0; i < n; i++) {
14         int count = 1;
15         if (arr[i] != -1) {
16             for (int j = i + 1; j < n; j++) {
17                 if (arr[i] == arr[j]) {
18                     count++;
19                     arr[j] = -1;
20                 }
21             }
22             printf("%d - %d\n", arr[i], count);
23         }
24     }
25
26     return 0;
27 }
28
```

Example Output:



The screenshot shows a Windows command prompt window with the following text:

```
demo.c X
#include
int
Enter the number of elements: 10
Enter the elements:
1 2 3 4 1 2 3 4 5 3
Element - Frequency
1 - 2
2 - 2
3 - 3
4 - 2
5 - 1

Process returned 0 (0x0)   execution time : 14.113 s
Press any key to continue.
```

5. Find Maximum and Minimum

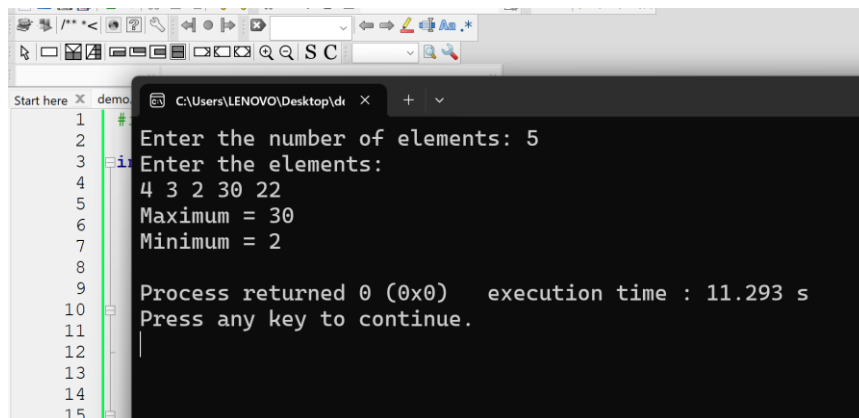
Description:

- Find the maximum and minimum elements in the array.

Code:

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

Example Output:



The screenshot shows a Windows command prompt window with a dark background. The program is running and has prompted the user to enter the number of elements (5) and the elements themselves (4 3 2 30 22). The output shows the maximum value is 30 and the minimum value is 2. The process returned 0 (0x0) and the execution time was 11.293 s. The prompt asks the user to press any key to continue.

```
1  #
2
3  Enter the number of elements: 5
4  Enter the elements:
5  4 3 2 30 22
6  Maximum = 30
7  Minimum = 2
8
9
10 Process returned 0 (0x0)   execution time : 11.293 s
11 Press any key to continue.
12
13
14
15
```

6. Sort the Array

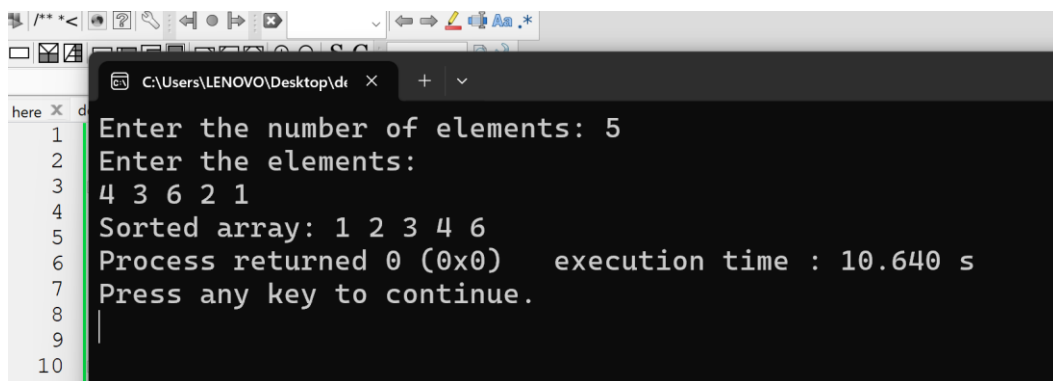
Description:

- Sort the array in ascending order.

Code:

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

Example Output:



```
C:\Users\LENOVO\Desktop\dt > Enter the number of elements: 5
Enter the elements:
4 3 6 2 1
Sorted array: 1 2 3 4 6
Process returned 0 (0x0)    execution time : 10.640 s
Press any key to continue.
```