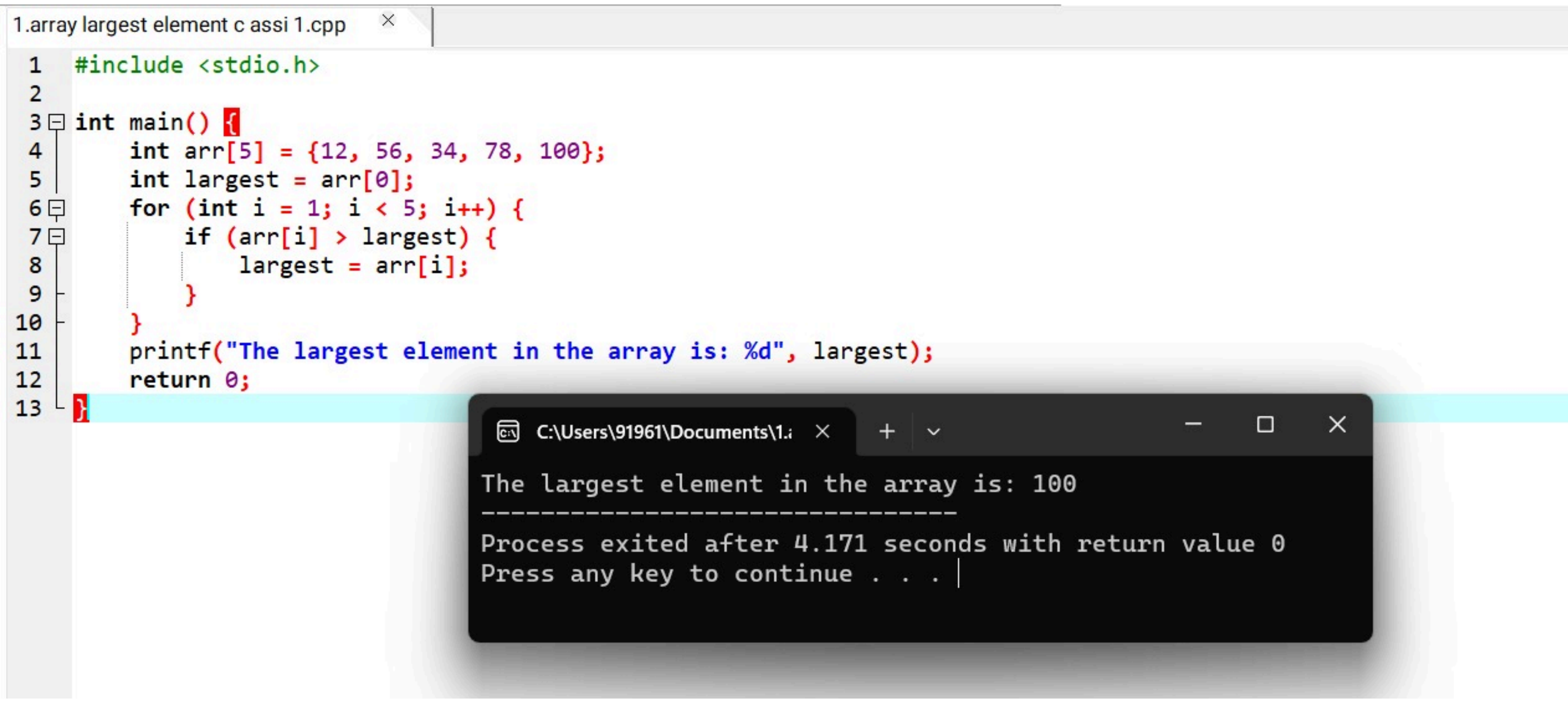
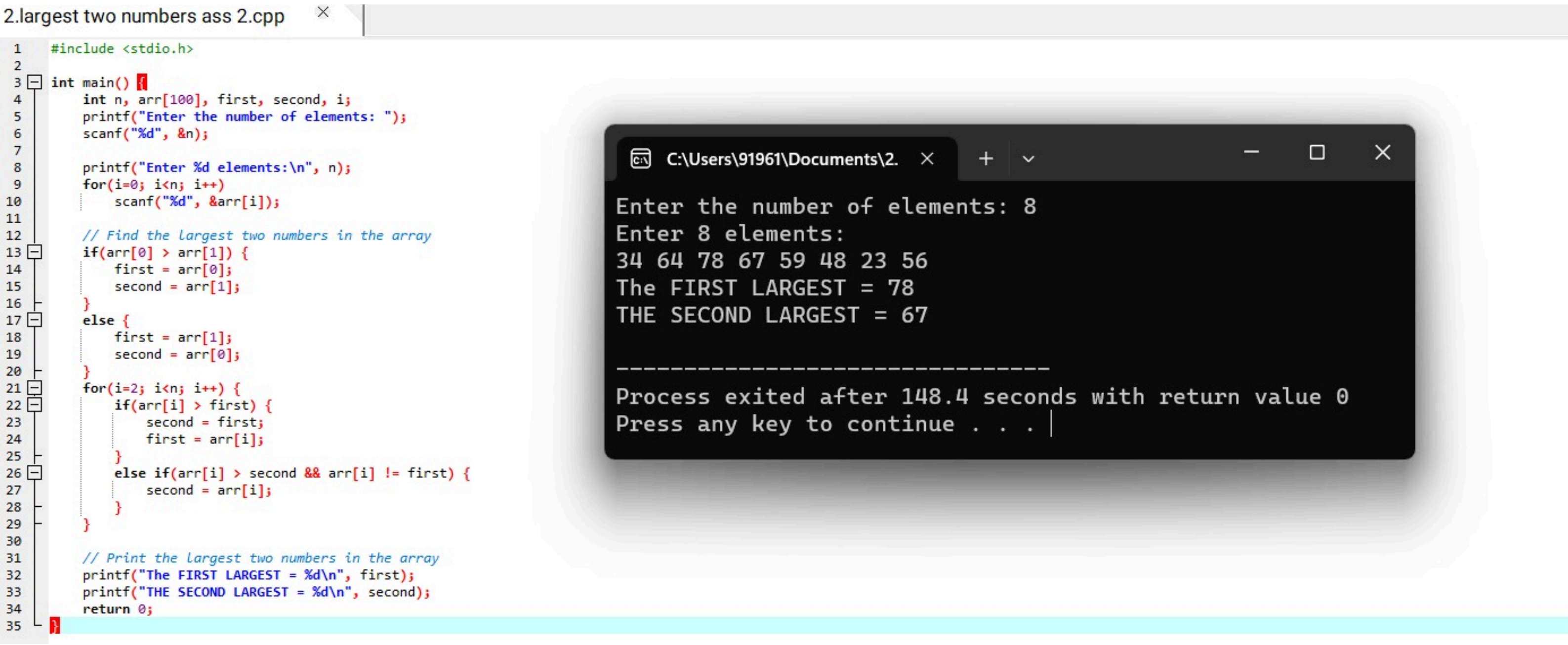


1.array largest element c assi 1.cpp
C:\Users\91961\Documents\1.a

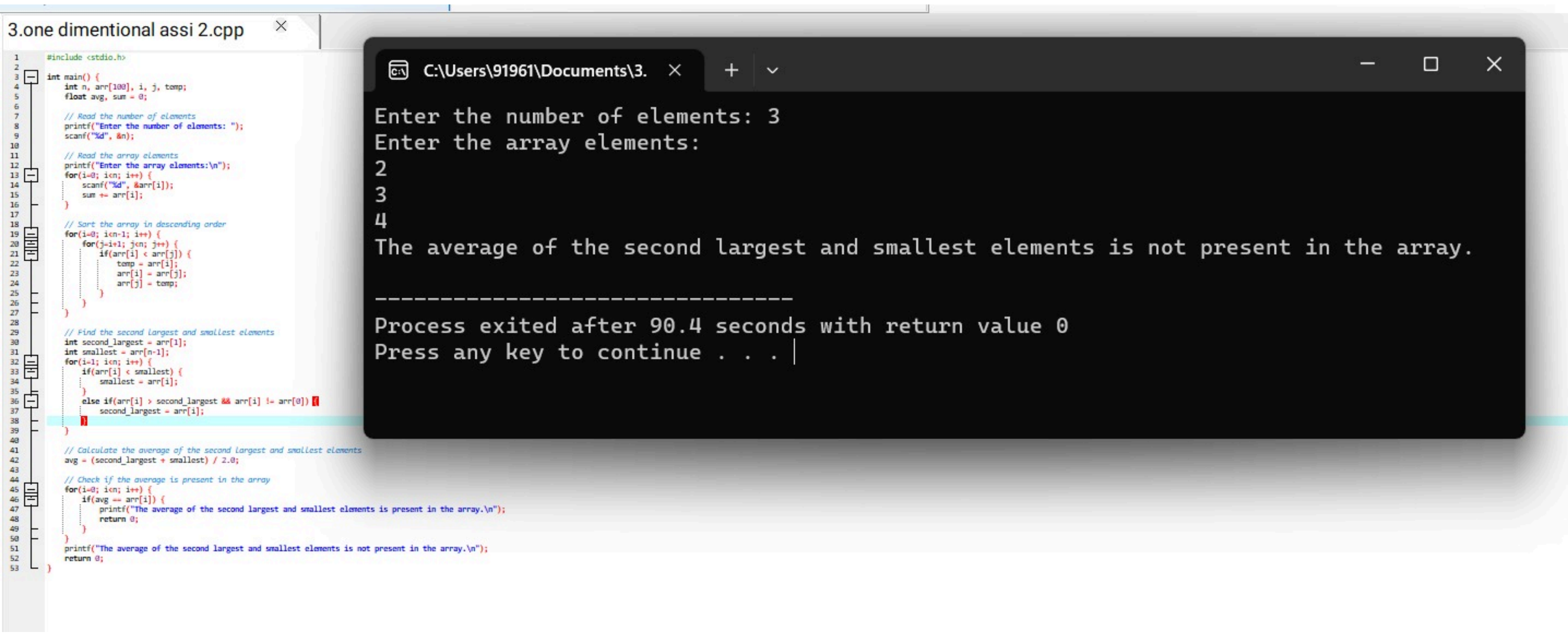
```c
#include <stdio.h>

int main() {
    int arr[5] = {12, 56, 34, 78, 100};
    int largest = arr[0];
    for (int i = 1; i < 5; i++) {
        if (arr[i] > largest) {
            largest = arr[i];
        }
    }
    printf("The largest element in the array is: %d", largest);
    return 0;
}
```

```
The largest element in the array is: 100
--------------------------------
Process exited after 4.171 seconds with return value 0
Press any key to continue . . .
```



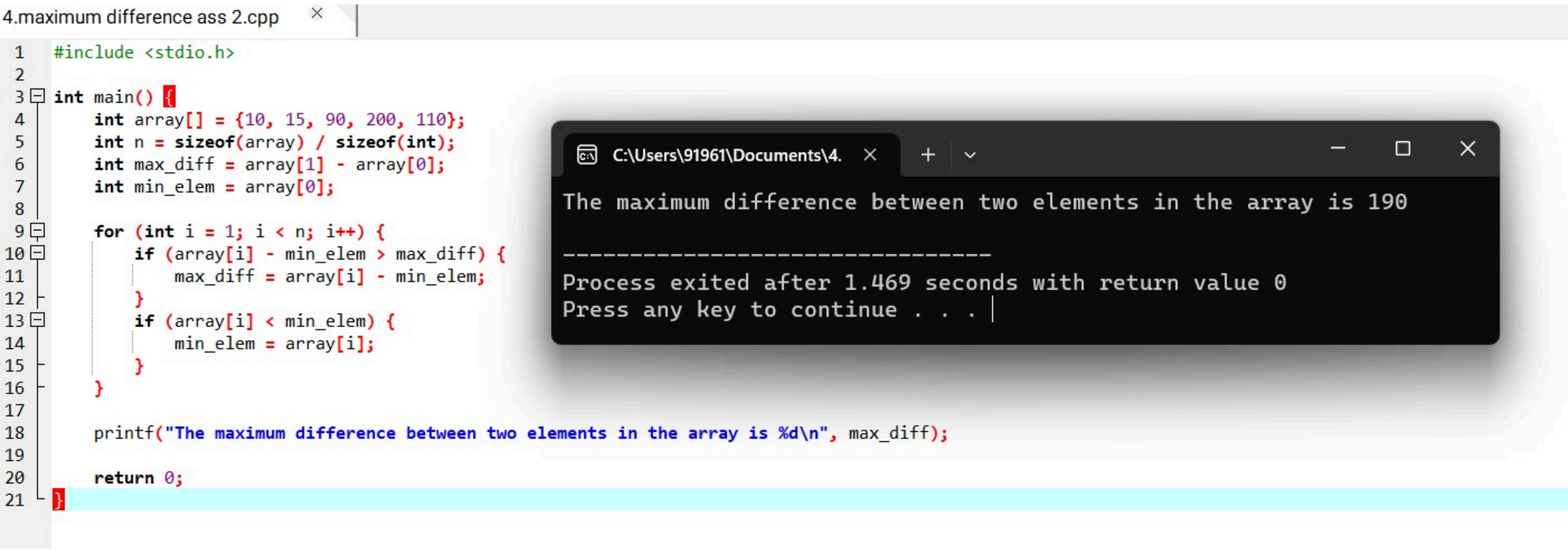
```c
#include <stdio.h>

int main() {
    int n, arr[100], first, second, i;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements:\n", n);
    for(i=0; i<n; i++)
        scanf("%d", &arr[i]);

    // Find the largest two numbers in the array
    if(arr[0] > arr[1]) {
        first = arr[0];
        second = arr[1];
    }
    else {
        first = arr[1];
        second = arr[0];
    }
    for(i=2; i<n; i++) {
        if(arr[i] > first) {
            second = first;
            first = arr[i];
        }
        else if(arr[i] > second && arr[i] != first) {
            second = arr[i];
        }
    }

    // Print the largest two numbers in the array
    printf("The FIRST LARGEST = %d\n", first);
    printf("THE SECOND LARGEST = %d\n", second);
    return 0;
}
```

```
Enter the number of elements: 8
Enter 8 elements:
34 64 78 67 59 48 23 56
The FIRST LARGEST = 78
THE SECOND LARGEST = 67

--------------------------------
Process exited after 148.4 seconds with return value 0
Press any key to continue . . .
```



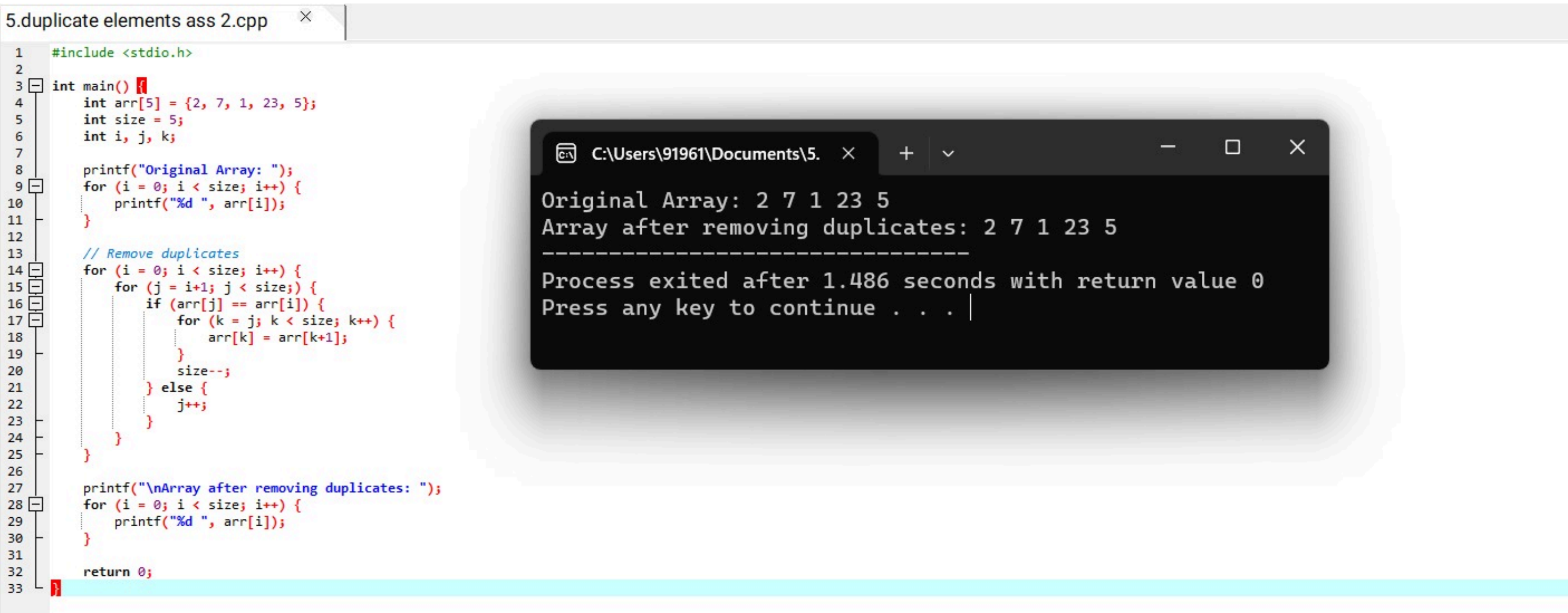
```c
#include <stdio.h>

int main() {
    int n, arr[100], i, j, temp;
    float avg, sum = 0;

    // Read the number of elements
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    // Read the array elements
    printf("Enter the array elements:\n");
    for(i=0; i<n; i++) {
        scanf("%d", &arr[i]);
        sum += arr[i];
    }

    // Sort the array in descending order
    for(i=0; i<n-1; i++) {
        for(j=i+1; j<n; j++) {
            if(arr[i] < arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    // Find the second largest and smallest elements
    int second_largest = arr[1];
    int smallest = arr[n-1];
    for(i=1; i<n; i++) {
        if(arr[i] < smallest) {
            smallest = arr[i];
        }
        else if(arr[i] > second_largest && arr[i] != arr[0]) {
            second_largest = arr[i];
        }
    }

    // Calculate the average of the second largest and smallest elements
    avg = (second_largest + smallest) / 2.0;

    // Check if the average is present in the array
    for(i=0; i<n; i++) {
        if(avg == arr[i]) {
            printf("The average of the second largest and smallest elements is present in the array.\n");
            return 0;
        }
    }
    printf("The average of the second largest and smallest elements is not present in the array.\n");
    return 0;
}
```

```
Enter the number of elements: 3
Enter the array elements:
2
3
4
The average of the second largest and smallest elements is not present in the array.

--------------------------------
Process exited after 90.4 seconds with return value 0
Press any key to continue . . .
```



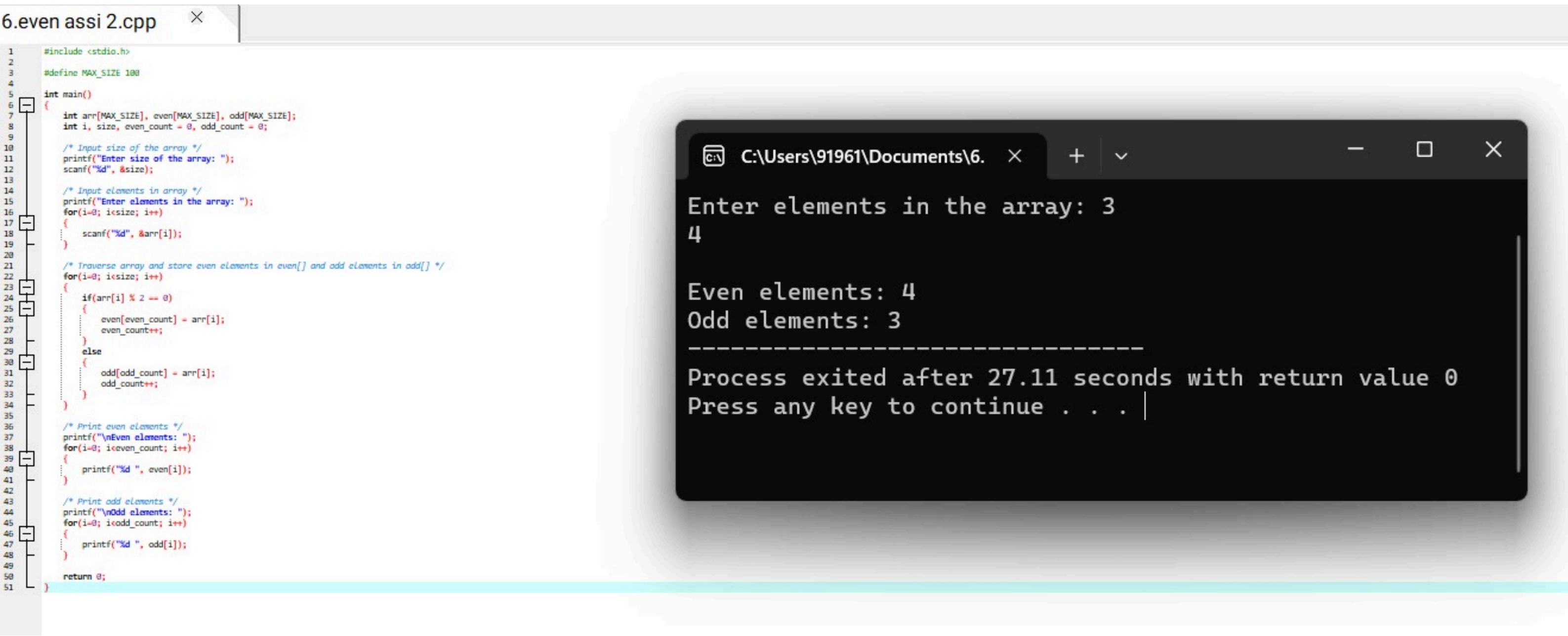
```c
#include <stdio.h>

int main() {
    int array[] = {10, 15, 90, 200, 110};
    int n = sizeof(array) / sizeof(int);
    int max_diff = array[1] - array[0];
    int min_elem = array[0];

    for (int i = 1; i < n; i++) {
        if (array[i] - min_elem > max_diff) {
            max_diff = array[i] - min_elem;
        }
        if (array[i] < min_elem) {
            min_elem = array[i];
        }
    }

    printf("The maximum difference between two elements in the array is %d\n", max_diff);

    return 0;
}
```

```
The maximum difference between two elements in the array is 190

--------------------------------
Process exited after 1.469 seconds with return value 0
Press any key to continue . . .
```



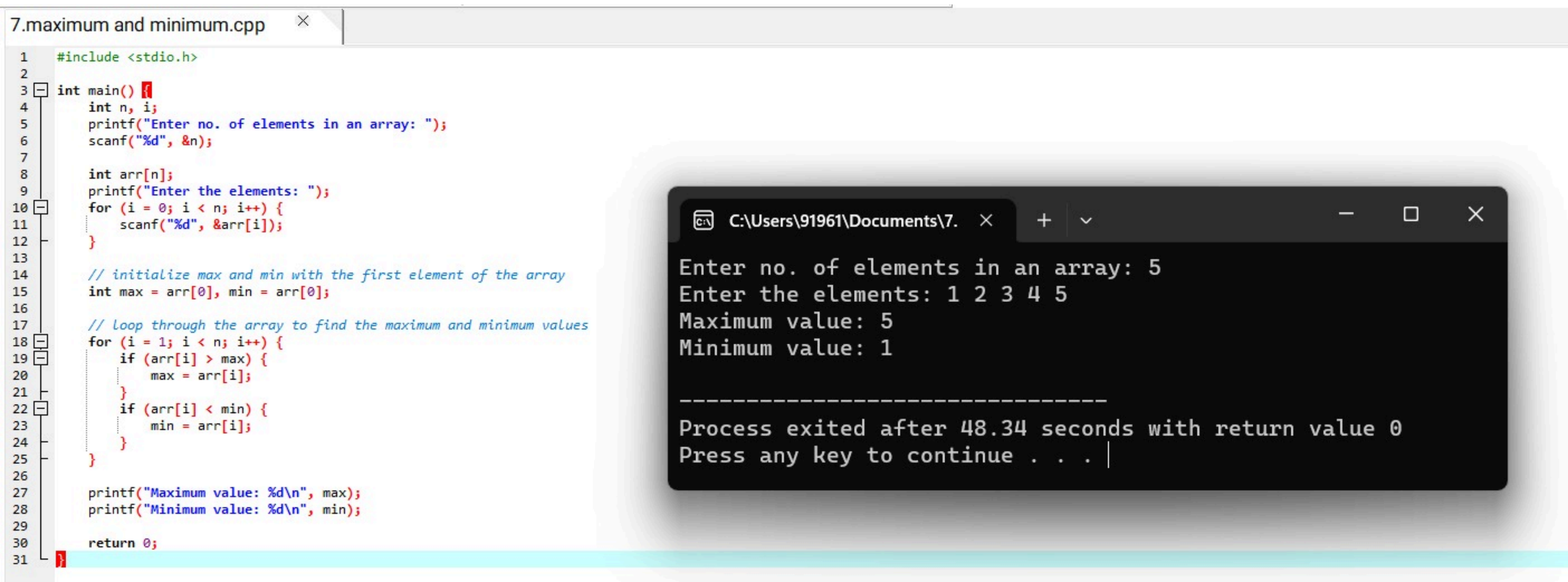
```c
#include <stdio.h>

int main() {
    int arr[5] = {2, 7, 1, 23, 5};
    int size = 5;
    int i, j, k;

    printf("Original Array: ");
    for (i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }

    // Remove duplicates
    for (i = 0; i < size; i++) {
        for (j = i+1; j < size;) {
            if (arr[j] == arr[i]) {
                for (k = j; k < size; k++) {
                    arr[k] = arr[k+1];
                }
                size--;
            } else {
                j++;
            }
        }
    }

    printf("\nArray after removing duplicates: ");
    for (i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

```
Original Array: 2 7 1 23 5
Array after removing duplicates: 2 7 1 23 5
--------------------------------
Process exited after 1.486 seconds with return value 0
Press any key to continue . . .
```



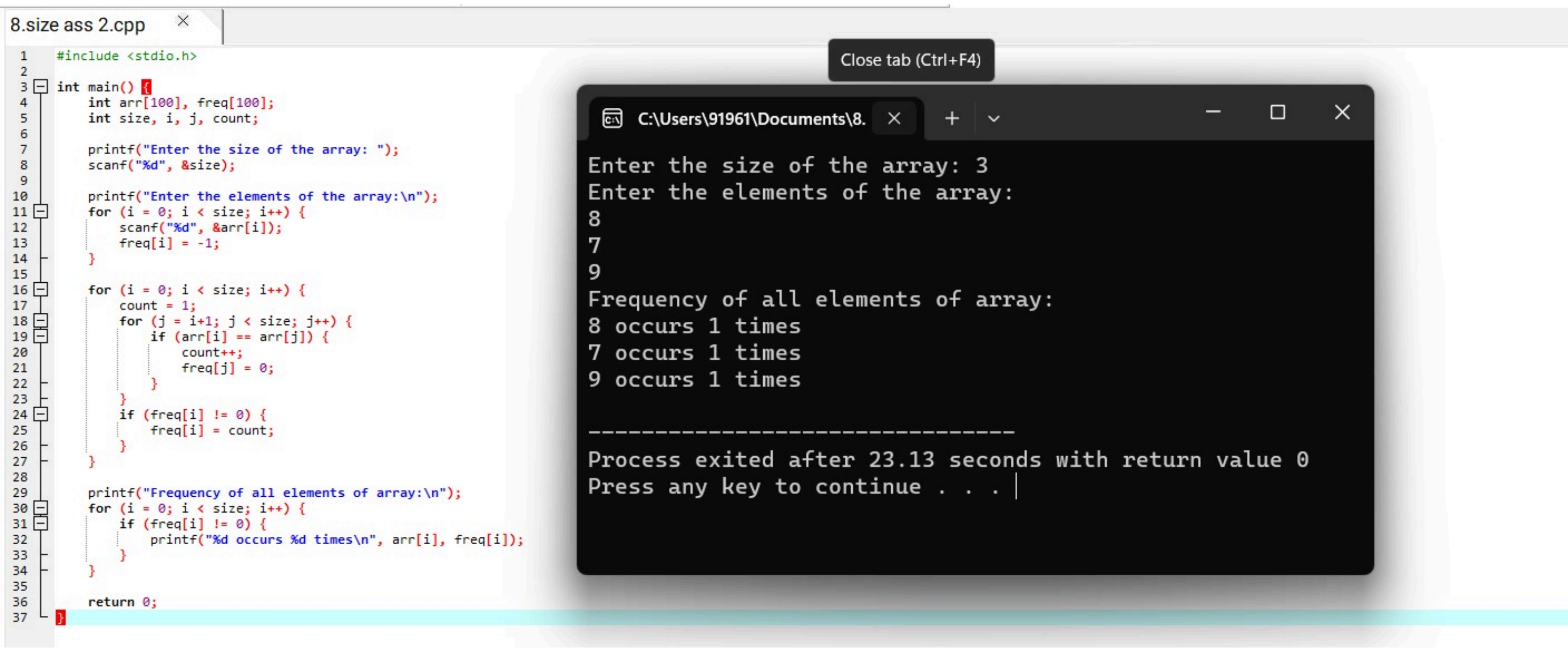
```c
#include <stdio.h>

#define MAX_SIZE 100

int main()
{
    int arr[MAX_SIZE], even[MAX_SIZE], odd[MAX_SIZE];
    int i, size, even_count = 0, odd_count = 0;

    /* Input size of the array */
    printf("Enter size of the array: ");
    scanf("%d", &size);

    /* Input elements in array */
    printf("Enter elements in the array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }

    /* Traverse array and store even elements in even[] and odd elements in odd[] */
    for(i=0; i<size; i++)
    {
        if(arr[i] % 2 == 0)
        {
            even[even_count] = arr[i];
            even_count++;
        }
        else
        {
            odd[odd_count] = arr[i];
            odd_count++;
        }
    }

    /* Print even elements */
    printf("\nEven elements: ");
    for(i=0; i<even_count; i++)
    {
        printf("%d ", even[i]);
    }

    /* Print odd elements */
    printf("\nOdd elements: ");
    for(i=0; i<odd_count; i++)
    {
        printf("%d ", odd[i]);
    }

    return 0;
}
```

```
Enter elements in the array: 3
4

Even elements: 4
Odd elements: 3
--------------------------------
Process exited after 27.11 seconds with return value 0
Press any key to continue . . .
```



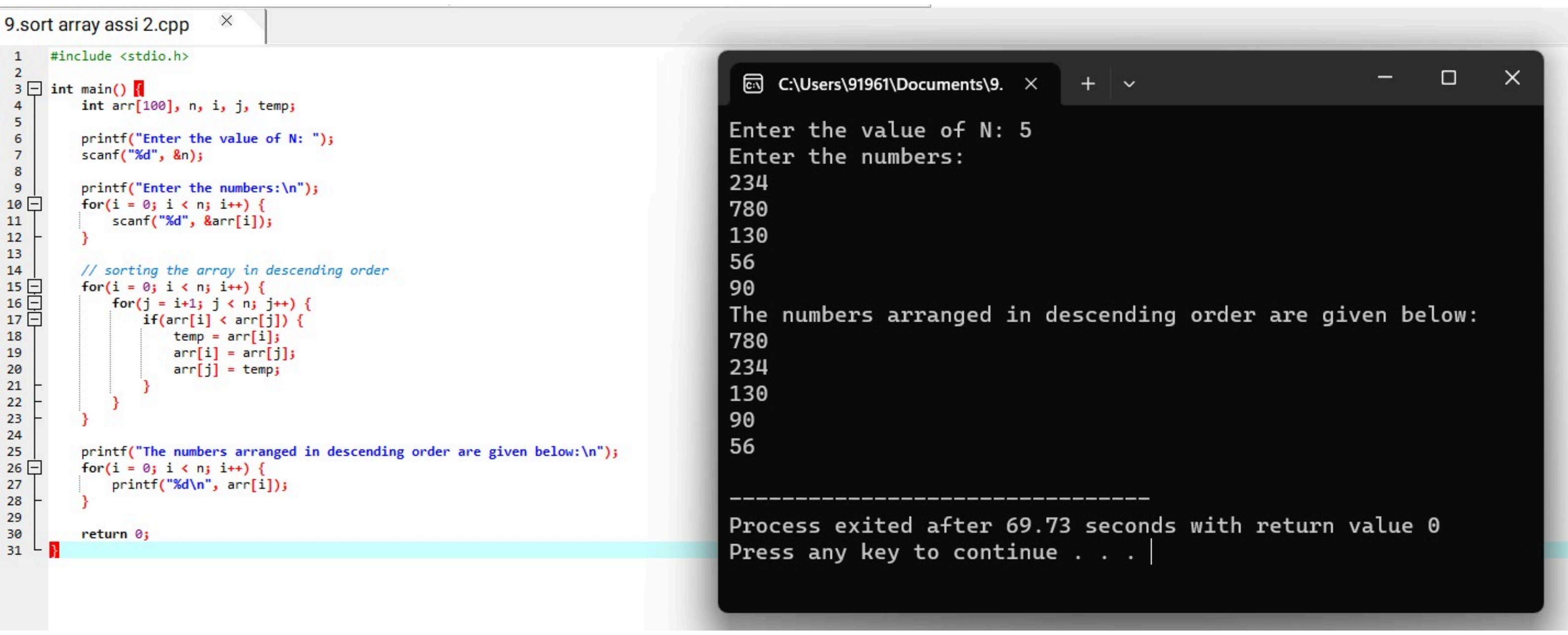
```c
#include <stdio.h>

int main() {
    int n, i;
    printf("Enter no. of elements in an array: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter the elements: ");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // initialize max and min with the first element of the array
    int max = arr[0], min = arr[0];

    // loop through the array to find the maximum and minimum values
    for (i = 1; i < n; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
        if (arr[i] < min) {
            min = arr[i];
        }
    }

    printf("Maximum value: %d\n", max);
    printf("Minimum value: %d\n", min);

    return 0;
}
```

```
Enter no. of elements in an array: 5
Enter the elements: 1 2 3 4 5
Maximum value: 5
Minimum value: 1

--------------------------------
Process exited after 48.34 seconds with return value 0
Press any key to continue . . .
```



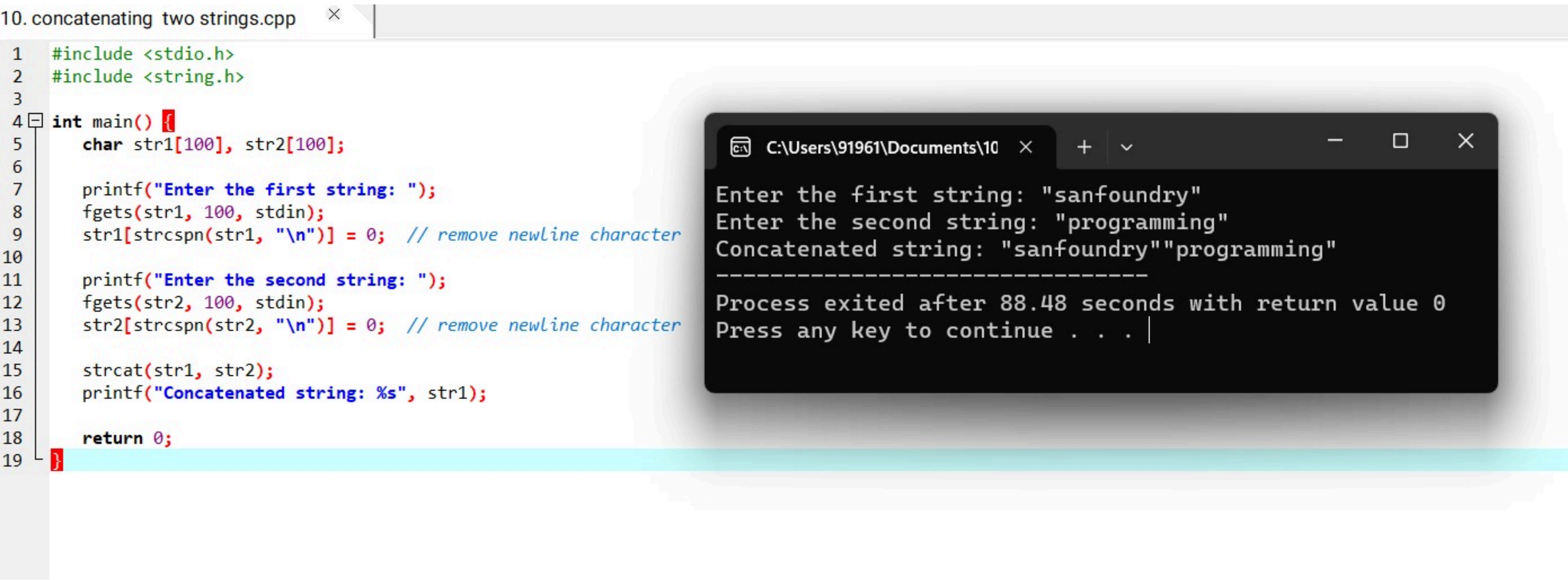
```
#include <stdio.h>

int main() {
    int arr[100], freq[100];
    int size, i, j, count;

    printf("Enter the size of the array: ");
    scanf("%d", &size);

    printf("Enter the elements of the array:\n");
    for (i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
        freq[i] = -1;
    }

    for (i = 0; i < size; i++) {
        count = 1;
        for (j = i+1; j < size; j++) {
            if (arr[i] == arr[j]) {
                count++;
                freq[j] = 0;
            }
        }
        if (freq[i] != 0) {
            freq[i] = count;
        }
    }

    printf("Frequency of all elements of array:\n");
    for (i = 0; i < size; i++) {
        if (freq[i] != 0) {
            printf("%d occurs %d times\n", arr[i], freq[i]);
        }
    }

    return 0;
}
```

```
Enter the size of the array: 3
Enter the elements of the array:
8
7
9
Frequency of all elements of array:
8 occurs 1 times
7 occurs 1 times
9 occurs 1 times

--------------------------------
Process exited after 23.13 seconds with return value 0
Press any key to continue . . .
```



```c
#include <stdio.h>

int main() {
    int arr[100], n, i, j, temp;

    printf("Enter the value of N: ");
    scanf("%d", &n);

    printf("Enter the numbers:\n");
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // sorting the array in descending order
    for(i = 0; i < n; i++) {
        for(j = i+1; j < n; j++) {
            if(arr[i] < arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    printf("The numbers arranged in descending order are given below:\n");
    for(i = 0; i < n; i++) {
        printf("%d\n", arr[i]);
    }

    return 0;
}
```

```
Enter the value of N: 5
Enter the numbers:
234
780
130
56
90
The numbers arranged in descending order are given below:
780
234
130
90
56

--------------------------------
Process exited after 69.73 seconds with return value 0
Press any key to continue . . .
```


10. concatenating two strings.cpp
C:\Users\91961\Documents\10

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

int main() {
    char str1[100], str2[100];

    printf("Enter the first string: ");
    fgets(str1, 100, stdin);
    str1[strcspn(str1, "\n")] = 0;  // remove newline character

    printf("Enter the second string: ");
    fgets(str2, 100, stdin);
    str2[strcspn(str2, "\n")] = 0;  // remove newline character

    strcat(str1, str2);
    printf("Concatenated string: %s", str1);

    return 0;
}
```

```
Enter the first string: "sanfoundry"
Enter the second string: "programming"
Concatenated string: "sanfoundry""programming"
--------------------------------
Process exited after 88.48 seconds with return value 0
Press any key to continue . . .
```