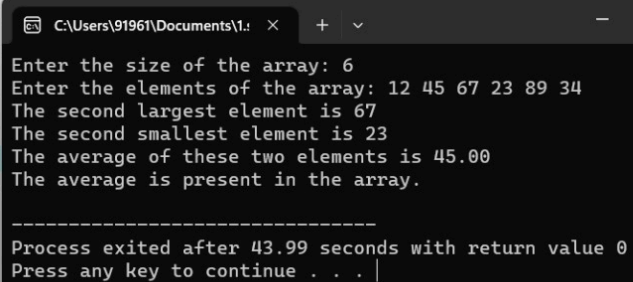


1.second largest and smallest elements.cpp

```
1 #include <stdio.h>
2 int main() {
3     int n, i, temp;
4     float avg;
5     printf("Enter the size of the array: ");
6     scanf("%d", &n);
7     int arr[n];
8     printf("Enter the elements of the array: ");
9     for (i = 0; i < n; i++) {
10         scanf("%d", &arr[i]);
11     }
12     // sorting the array in descending order
13     for (i = 0; i < n; i++) {
14         for (int j = i + 1; j < n; j++) {
15             if (arr[i] < arr[j]) {
16                 temp = arr[i];
17                 arr[i] = arr[j];
18                 arr[j] = temp;
19             }
20         }
21     }
22     // finding the second largest and smallest elements
23     int second_largest = arr[1];
24     int second_smallest = arr[n-2];
```



```
C:\Users\91961\Documents\1. x + -
Enter the size of the array: 6
Enter the elements of the array: 12 45 67 23 89 34
The second largest element is 67
The second smallest element is 23
The average of these two elements is 45.00
The average is present in the array.

-----
Process exited after 43.99 seconds with return value 0
Press any key to continue . . .
```

1.second largest and smallest elements.cpp

```
19     }
20 }
21 }
22 // finding the second largest and smallest elements
23 int second_largest = arr[1];
24 int second_smallest = arr[n-2];
25 avg = (float)(second_largest + second_smallest)/2;
26 printf("The second largest element is %d\n", second_largest);
27 printf("The second smallest element is %d\n", second_smallest);
28 printf("The average of these two elements is %.2f\n", avg);
29 // checking if the average is present in the array
30 for (i = 0; i < n; i++) {
31     if (arr[i] == avg) {
32         printf("The average is present in the array.\n");
33         break;
34     }
35 }
36 if (i == n) {
37     printf("The average is not present in the array.\n");
38 }
39 return 0;
40 }
41
```

```
C:\Users\91961\Documents\1. x + v - □ ×
Enter the size of the array: 6
Enter the elements of the array: 12 45 67 23 89 34
The second largest element is 67
The second smallest element is 23
The average of these two elements is 45.00
The average is present in the array.

-----
Process exited after 43.99 seconds with return value 0
Press any key to continue . . .
```

```

1  #include <stdio.h>
2
3  int main() {
4      int size, index, i;
5      printf("Enter the size of the array: ");
6      scanf("%d", &size);
7      int arr[size];
8
9      // Taking input of array
10     printf("Enter the elements of the array:\n");
11     for (i = 0; i < size; i++) {
12         scanf("%d", &arr[i]);
13     }
14
15     // Taking input of index to be deleted
16     printf("Enter the index of the element to be deleted: ");
17     scanf("%d", &index);
18
19     // Deleting the element by shifting all elements after
20     for (i = index; i < size - 1; i++) {
21         arr[i] = arr[i+1];
22     }
23     size--;
24 }

```

Sources Compile Log Debug Find Results Console Close

Errors: 0

```

C:\Users\91961\Documents\2. x + v
Enter the size of the array: 5
Enter the elements of the array:
8 4 9 6 2
Enter the index of the element to be deleted: 3
The updated array is:
8 4 9 2
-----
Process exited after 33.91 seconds with return value 0
Press any key to continue . . . |

```

2.delete an element in an array.cpp

```
10 printf("Enter the elements of the array:\n");
11 for (i = 0; i < size; i++) {
12     scanf("%d", &arr[i]);
13 }
14
15 // Taking input of index to be deleted
16 printf("Enter the index of the element to be deleted: ");
17 scanf("%d", &index);
18
19 // Deleting the element by shifting all elements after it to the left
20 for (i = index; i < size - 1; i++) {
21     arr[i] = arr[i+1];
22 }
23 size--;
24
25 // Printing the updated array
26 printf("The updated array is:\n");
27 for (i = 0; i < size; i++) {
28     printf("%d ", arr[i]);
29 }
30
31 return 0;
32 }
```

C:\Users\91961\Documents\2. × +

```
Enter the size of the array: 5
Enter the elements of the array: 8 4 9 6 2
Enter the index of the element to be deleted: 2
The updated array is:
8 4 9 2
```

```
-----
Process exited after 33.91 seconds
Press any key to continue . .
```

3.duplicate element.cpp

```
1 #include <stdio.h>
2
3 int main() {
4     int arr[] = {1, 2, 4, 5, 4, 2, 7, 5};
5     int n = sizeof(arr) / sizeof(arr[0]);
6     int i, j, k;
7
8     // Loop through each element in the array
9     for (i = 0; i < n; i++) {
10        // Check if the current element is a duplicate
11        for (j = i + 1; j < n; j++) {
12            if (arr[j] == arr[i]) {
13                // If it is a duplicate, shift all the elements to the left
14                for (k = j; k < n; k++) {
15                    arr[k] = arr[k + 1];
16                }
17                n--; // Decrement the size of the array
18            } else {
19                j++; // Move to the next element
20            }
21        }
22    }
23
24    // Print the resulting array
25    printf("Resultant Array after removing duplicates: ");
26    for (i = 0; i < n; i++) {
27        printf("%d ", arr[i]);
28    }
29    printf("\n");
30
31    return 0;
32 }
```

C:\Users\91961\Documents\3. x + v - □ ×

Resultant Array after removing duplicates: 1 2 4 5 7

Process exited after 3.02 seconds with return value 0

Press any key to continue . . . |

4.merge.cpp

```
1 #include <stdio.h>
2
3 void mergeArrays(int arr1[], int arr2[], int n1, int n2, int mergedArr[]) {
4     int i = 0, j = 0, k = 0;
5
6     while (i < n1 && j < n2) {
7         if (arr1[i] <= arr2[j]) {
8             mergedArr[k++] = arr1[i++];
9         } else {
10             mergedArr[k++] = arr2[j++];
11         }
12     }
13
14     while (i < n1) {
15         mergedArr[k++] = arr1[i++];
16     }
17
18     while (j < n2) {
19         mergedArr[k++] = arr2[j++];
20     }
21 }
22
23 int main() {
24     int arr1[] = {1, 3, 5, 7, 9};
25     int arr2[] = {2, 4, 6, 8, 10};
26     int n1 = sizeof(arr1) / sizeof(arr1[0]);
27     int n2 = sizeof(arr2) / sizeof(arr2[0]);
28     int mergedArr[n1 + n2];
29
30     mergeArrays(arr1, arr2, n1, n2, mergedArr);
31
32     printf("Merged Array:\n");
33     for (int i = 0; i < n1 + n2; i++) {
34         printf("%d", mergedArr[i]);
35         if (i < n1 + n2 - 1) {
36             printf(" ");
37         }
38     }
39     printf("\n");
40
41     return 0;
42 }
```

C:\Users\91961\Documents\4. × + ▾

Merged Array:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Process exited after 2.841 seconds with return value 0
Press any key to continue . . . |

5.diagonal.cpp

```
1  #include <stdio.h>
2  int main()
3  {
4      int matrix[3][3]={ {1,2,3},{4,5,6},{7,8,9}};
5      int i, diagonal_sum=0;
6      printf("diagonal elements are");
7      for(i=0;i<3;i++){
8          printf("    %d",matrix[i][i]);
9          diagonal_sum += matrix[i][i];
10     }
11     printf("\nsum of diagonal elements =%d",diagonal_sum);
12     return 0;
13 }
```

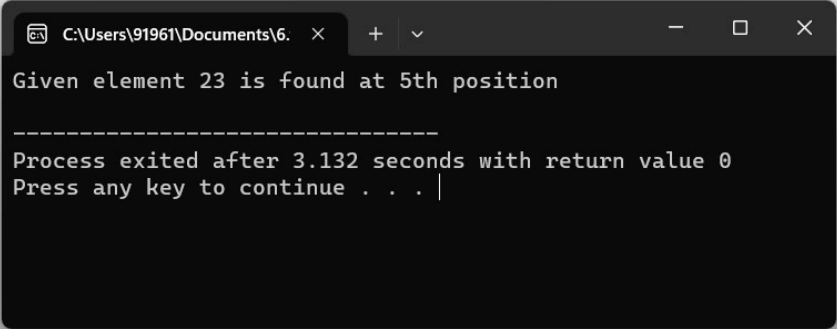
C:\Users\91961\Documents\5. x + v

diagonal elements are 1 5 9
sum of diagonal elements =15

Process exited after 2.173 seconds with return value 0
Press any key to continue . . . |

6.track of element.cpp

```
1 #include <stdio.h>
2
3 int main()
4 // input array
5 int arr[] = {16, 18, 27, 16, 23, 21, 19};
6 int n = sizeof(arr) / sizeof(arr[0]);
7
8 // element to search
9 int x = 23;
10
11 // flag to keep track if element is found
12 int found = 0;
13
14 // Loop through each element in the array
15 for (int i = 0; i < n; i++) {
16 // check if current element is the element we're looking for
17 if (arr[i] == x) {
18 // if yes, set found flag to 1 and print its position
19 found = 1;
20 printf("Given element %d is found at %dth position\n", x, i+1);
21 break;
22 }
23
24 // if element is not found, print message
25 if (!found) {
26 printf("Given element %d is not found in the array\n", x);
27 }
28
29 return 0;
30
31
```



7.maximum and minimum.cpp

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n, i;
6     printf("Enter no. of elements in an array: ");
7     scanf("%d", &n);
8
9     int arr[n];
10    printf("Enter the elements: ");
11    for (i = 0; i < n; i++) {
12        scanf("%d", &arr[i]);
13    }
14
15    // initialize max and min with the first element of the array
16    int max = arr[0], min = arr[0];
17
18    // Loop through the array to find the maximum and minimum values
19    for (i = 1; i < n; i++) {
20        if (arr[i] > max) {
21            max = arr[i];
22        }
23        if (arr[i] < min) {
24            min = arr[i];
25        }
26    }
27
28    printf("Maximum value: %d\n", max);
29    printf("Minimum value: %d\n", min);
30
31    return 0;
}
```

```
C:\Users\91961\Documents\7. × + ▾
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 . . . |
```

8.calculate of length.cpp

```
1  #include <stdio.h>
2
3  int main() {
4      int arr[] = {16, 18, 27, 16, 23, 21, 19};
5      int n = sizeof(arr) / sizeof(arr[0]); // calculate the length of the array
6      int sum = 0;
7      float avg;
8
9      // Loop through the array to find the sum of all elements
10     for (int i = 0; i < n; i++) {
11         sum += arr[i];
12     }
13     // calculate the average
14     avg = (float) sum / n;
15     printf("Sum of all elements: %d\n", sum);
16     printf("Average of all elements: %.2f\n", avg); // print the average with two decimal places
17
18     return 0;
19 }
```

C:\Users\91961\Documents\8. x + v

Sum of all elements: 140
Average of all elements: 20.00

Process exited after 2.473 seconds with return value 0
Press any key to continue . . . |

9.string.cpp

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char statement[100];
6      int i, vowels = 0;
7
8      printf("Enter a statement: ");
9      fgets(statement, 100, stdin);
10
11     for (i = 0; i < strlen(statement); i++) {
12         if (statement[i] == 'a' || statement[i] == 'e' || statement[i] == 'i' ||
13             statement[i] == 'o' || statement[i] == 'u' || statement[i] == 'A' ||
14             statement[i] == 'E' || statement[i] == 'I' || statement[i] == 'O' ||
15             statement[i] == 'U') {
16             vowels++;
17         }
18     }
19
20     printf("Number of vowels in the statement: %d\n", vowels);
21
22     return 0;
23 }
```

```
C:\Users\91961\Documents\9.  x + v - □ x
Enter a statement: saveetha school of engineering
Number of vowels in the statement: 12

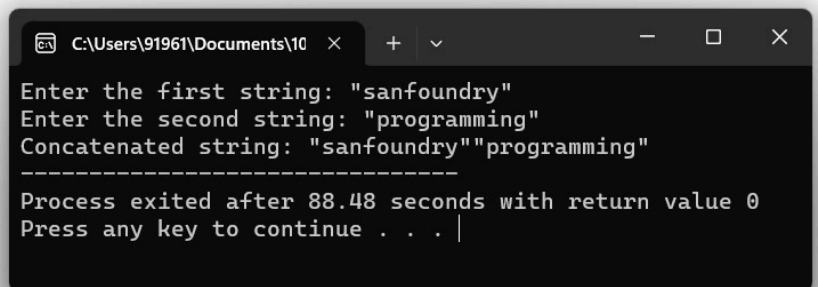
-----
Process exited after 41.81 seconds with return value 0
Press any key to continue . . . |
```

sources Compile Log Debug Find Results Console Close

Errors: 0

10. concatenating two strings.cpp

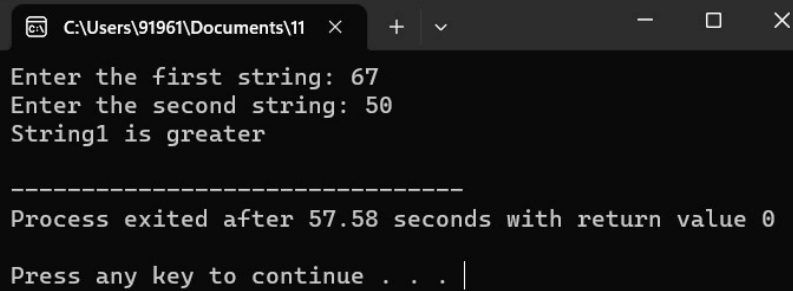
```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5     char str1[100], str2[100];
6
7     printf("Enter the first string: ");
8     fgets(str1, 100, stdin);
9     str1[strcspn(str1, "\n")] = 0; // remove newline character
10
11     printf("Enter the second string: ");
12     fgets(str2, 100, stdin);
13     str2[strcspn(str2, "\n")] = 0; // remove newline character
14
15     strcat(str1, str2);
16     printf("Concatenated string: %s", str1);
17
18     return 0;
19 }
```



```
C:\Users\91961\Documents\10
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 . . . |
```

11.string comparison in c.cpp

```
1
2 #include <stdio.h>
3
4 int main() {
5     char string1[50], string2[50];
6     int i, flag = 0;
7
8     printf("Enter the first string: ");
9     scanf("%s", string1);
10
11     printf("Enter the second string: ");
12     scanf("%s", string2);
13
14     // Comparing the strings character by character
15     for (i = 0; string1[i] != '\0' || string2[i] != '\0'; i++) {
16         if (string1[i] != string2[i]) {
17             flag = 1;
18             break;
19         }
20     }
21
22     if (flag == 0) {
23         printf("Both strings are equal\n");
24     } else if (string1[i] > string2[i]) {
25         printf("String1 is greater\n");
26     } else {
27         printf("String2 is greater\n");
28     }
29
30     return 0;
31 }
```



```
C:\Users\91961\Documents\11
Enter the first string: 67
Enter the second string: 50
String1 is greater

-----
Process exited after 57.58 seconds with return value 0
Press any key to continue . . . |
```

12.remove characters in c.cpp

```
1  #include <stdio.h>
2  #include <string.h>
3
4  void removeChars(char *str1, char *str2);
5
6  int main() {
7      char str1[100], str2[100];
8      printf("Enter the first string: ");
9      scanf("%s", str1);
10     printf("Enter the second string: ");
11     scanf("%s", str2);
12     removeChars(str1, str2);
13     printf("The second string after removing characters present in first string: %s\n", str2);
14     return 0;
15 }
16
17 void removeChars(char *str1, char *str2) {
18     int len1 = strlen(str1), len2 = strlen(str2), k = 0;
19     for (int i = 0; i < len2; i++) {
20         int j;
21         for (j = 0; j < len1; j++) {
22             if (str1[j] == str2[i]) {
23                 break;
24             }
25         }
26         if (j == len1) {
27             str2[k++] = str2[i];
28         }
29     }
30     str2[k] = '\0';
31 }
```

```
C:\Users\91961\Documents\12 x + v
Enter the first string: sanfoundary
Enter the second string: sanppppdry
The second string after removing characters present in first string: pppp
-----
Process exited after 71.93 seconds with return value 0
Press any key to continue . . . |
```

13.reverse string in c.cpp

```
1 #include <stdio.h>
2 #include <string.h>
3
4 void reverse(char str[]) {
5     int len = strlen(str);
6     for(int i=0; i<len/2; i++) {
7         char temp = str[i];
8         str[i] = str[len-i-1];
9         str[len-i-1] = temp;
10    }
11 }
12
13 int main() {
14     char str[100];
15     printf("Enter a string: ");
16     fgets(str, 100, stdin); // read string from user
17     reverse(str); // call the function to reverse the string
18     printf("Reversed string: %s", str); // print the reversed string
19     return 0;
20 }
```

C:\Users\91961\Documents\13

Enter a string: "hello"
Reversed string:
"olleh"

Process exited after 26.17 seconds with return value 0
Press any key to continue . . . |

14.length of the string.cpp

```
1  #include <stdio.h>
2
3  int main() {
4      char str[100];
5      int length = 0;
6
7      printf("Enter a string: ");
8      fgets(str, 100, stdin);
9
10     while (str[length] != '\0') {
11         length++;
12     }
13
14     printf("Length of string: %d\n", length);
15
16     return 0;
17 }
```

```
C:\Users\91961\Documents\14  ×  +  ▾  -  □  ×
Enter a string: I love programming. I love Codeforwin
Length of string: 38

-----
Process exited after 72.15 seconds with return value 0
Press any key to continue . . . |
```


15.remove all occurence.cpp

```
1  #include <stdio.h>
2
3  void remove_char(char *str, char c) {
4      int i, j;
5      for (i = 0, j = 0; str[i] != '\0'; i++) {
6          if (str[i] != c) {
7              str[j] = str[i];
8              j++;
9          }
10     }
11     str[j] = '\0';
12 }
13
14 int main() {
15     char str[100];
16     char c;
17
18     printf("Enter a string: ");
19     fgets(str, 100, stdin);
20
21     printf("Enter the character to remove: ");
22     scanf("%c", &c);
23
24     remove_char(str, c);
25
26     printf("Result: %s\n", str);
27
28     return 0;
29 }
```

```
C:\Users\91961\Documents\15 x + v - □ ×
Enter a string: I LOVE Programming.I Love Codeforwin
Enter the character to remove: 'I'
Result: I LOVE Programming.I Love Codeforwin

-----
Process exited after 83.68 seconds with return value 0
Press any key to continue . . . |
```

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4
5 void removeExtraSpaces(char *str) {
6     int i = 0, j = 0;
7     int flag = 0; // Flag to keep track of spaces
8
9     while (str[i]) {
10        if (isspace(str[i])) {
11            // If space is encountered
12            if (flag == 0) {
13                // If it is first space in a sequence of spaces
14                str[j++] = ' '; // Add single space to output
15                flag = 1; // Set flag to 1
16            }
17        } else {
18            // If non-space character is encountered
19            str[j++] = str[i]; // Add it to output
20            flag = 0; // Reset flag to 0
21        }
22        i++;
23    }
24    str[j] = '\0'; // Add null character at the end
25 }
26
27 int main() {
28     char str[100];
29
30     printf("Enter a string: ");
31     fgets(str, 100, stdin);
32
33     printf("Before removing extra spaces: %s\n", str);
34     removeExtraSpaces(str);
35
36     printf("After removing extra spaces: %s\n", str);
37
38     return 0;
39 }
40

```

```

C:\Users\91961\Documents\16 x + v - □ ×
Enter a string: Learn C programming at Cadeforwin
Before removing extra spaces: Learn C programming at Cadeforwin

After removing extra spaces: Learn C programming at Cadeforwin

-----
Process exited after 60.92 seconds with return value 0
Press any key to continue . . . |

```

17.trailing white trims.cpp

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4
5 void trimWhiteSpace(char *str) {
6     int len = strlen(str);
7     int start = 0, end = len - 1;
8
9     // Trim leading spaces
10    while (isspace(str[start])) {
11        start++;
12    }
13
14    // Trim trailing spaces
15    while (end >= start && isspace(str[end])) {
16        end--;
17    }
18
19    // Move the trimmed string to beginning of the input string
20    int i;
21    for (i = start; i <= end; i++) {
22        str[i - start] = str[i];
23    }
24    str[i - start] = '\0'; // Add null character at the end of the trimmed string
25
26
27 int main() {
28     char str[100];
29
30     printf("Enter a string: ");
31     fgets(str, 100, stdin);
32
33     printf("Before trimming white space characters: %s\n", str);
34     trimWhiteSpace(str);
35
36     printf("After trimming white space characters: %s\n", str);
37
38     return 0;
39 }
40
```

Sources Compile Log Debug Find Results

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\91961\Documents\17.trailing white trims.exe
- Output Size: 379.9482421875 KiB
- Compilation Time: 0.27s
```

```
C:\Users\91961\Documents\17
Enter a string: "Lost of leading and trailing spaces."
Before trimming white space characters: "Lost of leading and trailing spaces."

After trimming white space characters: "Lost of leading and trailing spaces."

-----
Process exited after 48.85 seconds with return value 0
Press any key to continue . . .
```