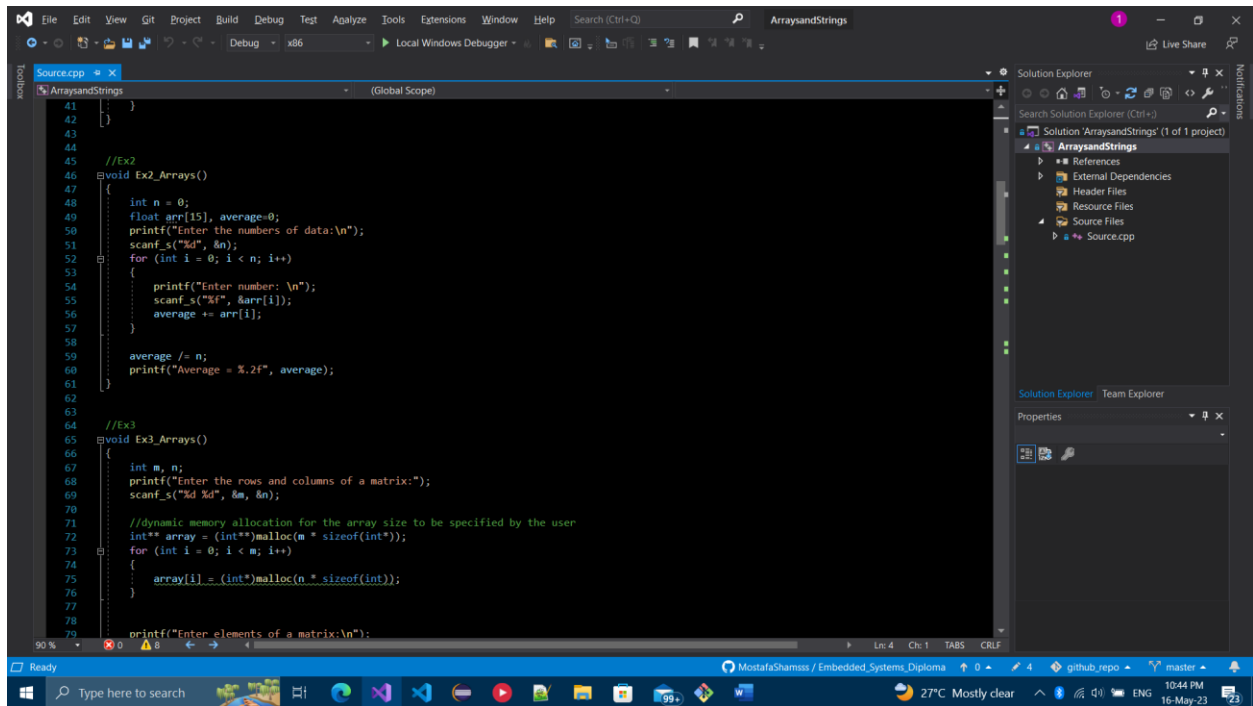


```
5 //***** Arrays *****/
6 //Ex1
7 void Ex1_Arrays()
8 {
9     const int n = 2;
10    float arr1[n][n], arr2[n][n];
11
12    printf("Enter the elements of the 1st matrix\n");
13    for (int i = 0; i < 2; i++)
14    {
15        for (int j = 0; j < 2; j++)
16        {
17            printf("Enter a%d%d:", i+1, j+1);
18            scanf_s("%f", &arr1[i][j]);
19        }
20    }
21
22    printf("Enter the elements of the 2nd matrix\n");
23    for (int i = 0; i < 2; i++)
24    {
25        for (int j = 0; j < 2; j++)
26        {
27            printf("Enter a%d%d:", i+1, j+1);
28            scanf_s("%f", &arr2[i][j]);
29        }
30    }
31
32    printf("Sum of Matrix:\n");
33    for (int i = 0; i < 2; i++)
34    {
35        printf("\n");
36        for (int j = 0; j < 2; j++)
37        {
38            printf("%1f\t", (arr1[i][j] + arr2[i][j]));
39        }
40    }
41 }
42
43
```



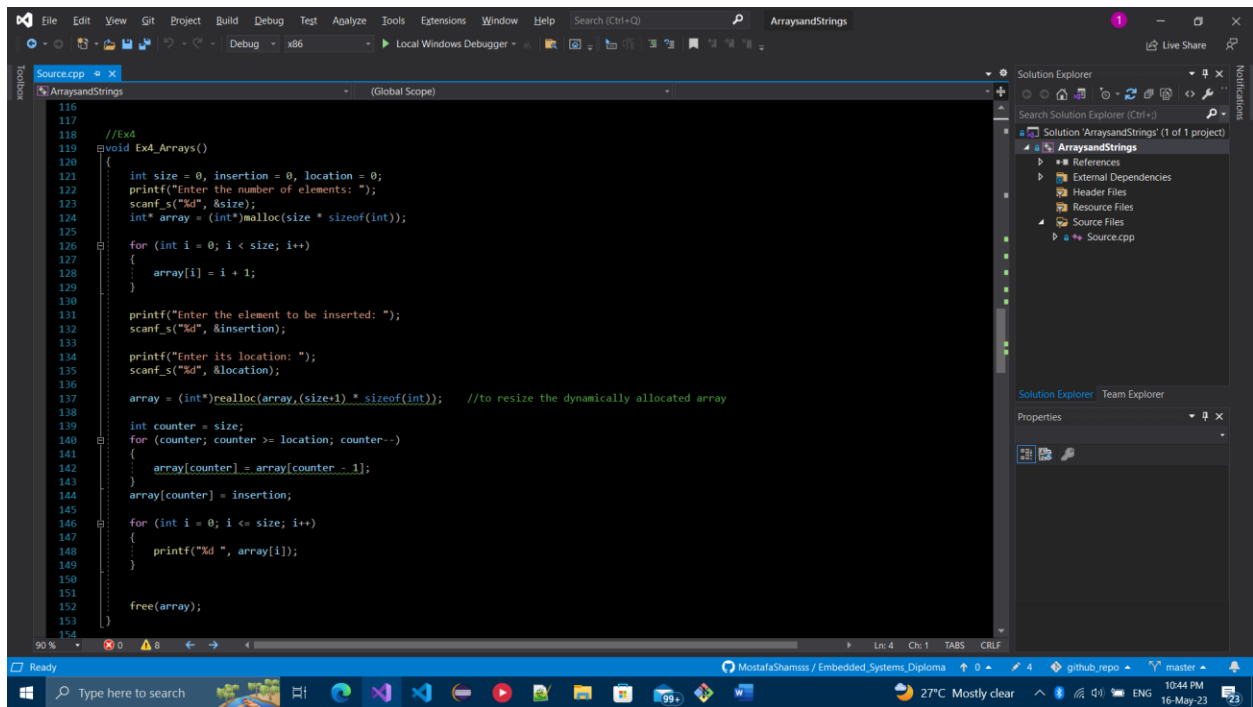
```
41 }
42 }
43
44 //Ex2
45 void Ex2_Arrays()
46 {
47     int n = 0;
48     float arr[15], average=0;
49     printf("Enter the numbers of data:\n");
50     scanf_s("%d", &n);
51     for (int i = 0; i < n; i++)
52     {
53         printf("Enter number: \n");
54         scanf_s("%f", &arr[i]);
55         average += arr[i];
56     }
57
58     average /= n;
59     printf("Average = %.2f", average);
60 }
61
62 //Ex3
63 void Ex3_Arrays()
64 {
65     int m, n;
66     printf("Enter the rows and columns of a matrix:");
67     scanf_s("%d %d", &m, &n);
68
69     //dynamic memory allocation for the array size to be specified by the user
70     int** array = (int**)malloc(m * sizeof(int*));
71     for (int i = 0; i < m; i++)
72     {
73         array[i] = (int*)malloc(n * sizeof(int));
74     }
75
76     printf("Enter elements of a matrix:\n");
77
78
79
```

This screenshot shows the first part of the C++ code in the 'Source.cpp' file. The code defines a function `Ex3_Arrays()` that takes two integers, `m` and `n`, as input. It prompts the user to enter the rows and columns of a matrix. The code then dynamically allocates memory for a 2D array `array` of size `m * sizeof(int)`. It uses nested loops to prompt the user to enter elements of the matrix and prints the entered matrix. Finally, it prints the transpose of the matrix. The code is written in a dark-themed editor with line numbers from 62 to 100. The Solution Explorer on the right shows the project structure for 'ArraysandStrings'.

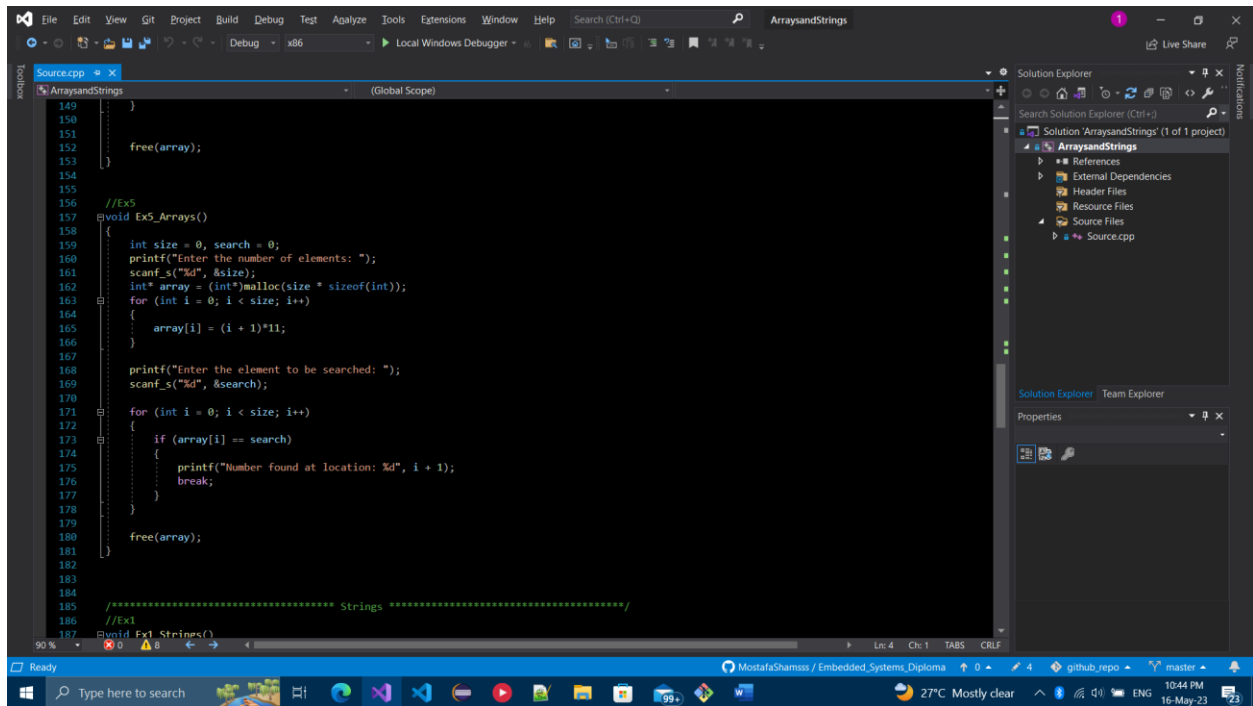
```
62 //Ex3
63
64 void Ex3_Arrays()
65 {
66     int m, n;
67     printf("Enter the rows and columns of a matrix:");
68     scanf_s("%d %d", &m, &n);
69
70     //dynamic memory allocation for the array size to be specified by the user
71     int** array = (int**)malloc(m * sizeof(int*));
72     for (int i = 0; i < m; i++)
73     {
74         array[i] = (int*)malloc(n * sizeof(int));
75     }
76
77     printf("Enter elements of a matrix:\n");
78     for (int i = 0; i < m; i++)
79     {
80         for (int j = 0; j < n; j++)
81         {
82             printf("Enter element a%d%d: ", i + 1, j + 1);
83             scanf_s("%d", &array[i][j]);
84         }
85     }
86
87     printf("Entered matrix:\n");
88     for (int i = 0; i < m; i++)
89     {
90         printf("\n");
91         for (int j = 0; j < n; j++)
92         {
93             printf("%d\t", array[i][j]);
94         }
95     }
96
97     printf("\nTranspose of a Matrix:");
98     for (int i = 0; i < n; i++)
99     {
100         for (int j = 0; j < m; j++)
101         {
102             printf("%d\t", array[j][i]);
103         }
104     }
105 }
```

This screenshot shows the second part of the C++ code in the 'Source.cpp' file. The code continues from the previous part, showing the memory deallocation for the 2D array `array` and the definition of a new function `Ex4_Arrays()`. The code is written in a dark-themed editor with line numbers from 101 to 121. The Solution Explorer on the right shows the project structure for 'ArraysandStrings'.

```
101 //free the memory allocated to the array
102 for (int i = 0; i < m; i++) {
103     free(array[i]);
104 }
105 free(array);
106
107 //Ex4
108 void Ex4_Arrays()
109 {
110     int size = 0, insertion = 0, location = 0;
111 }
```



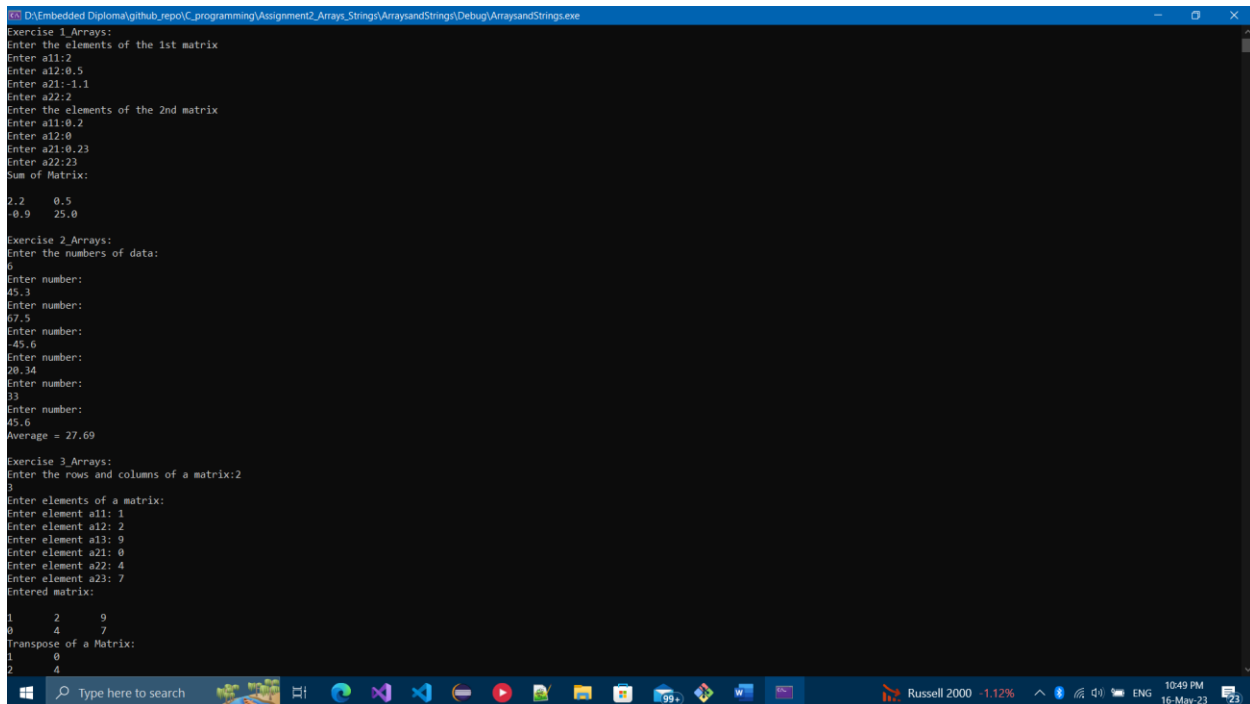
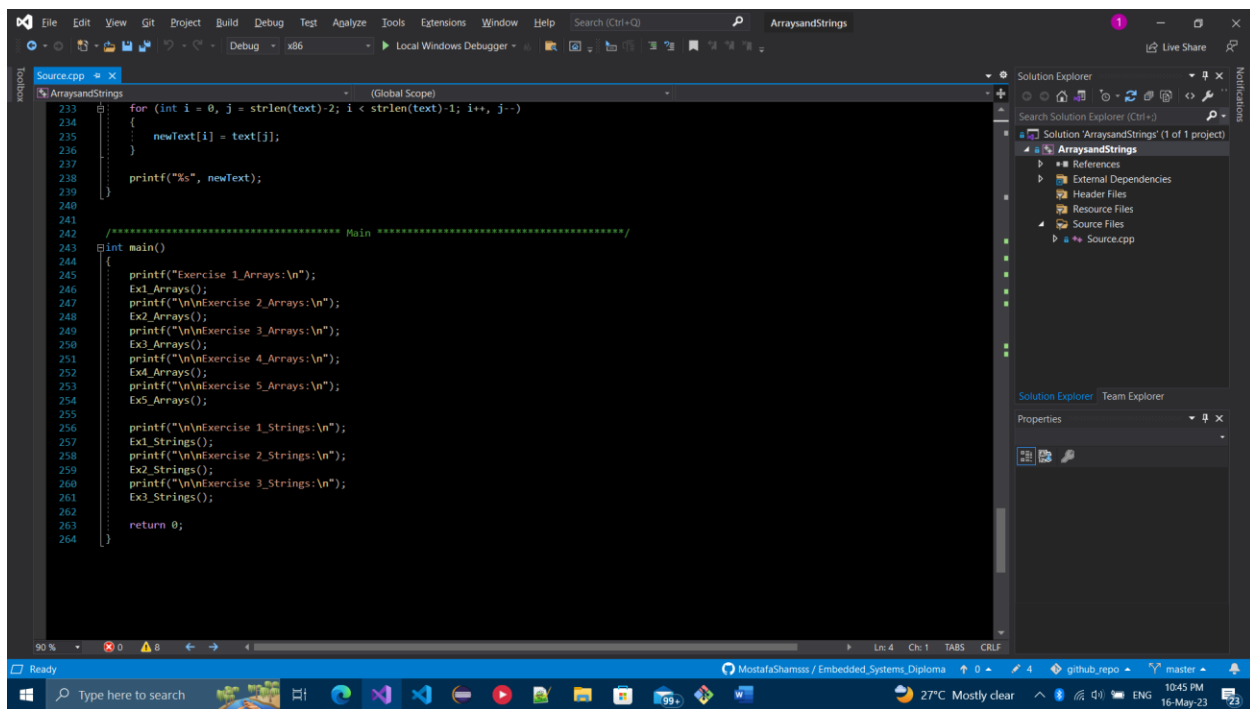
```
116
117
118 //Ex4
119 void Ex4_Arrays()
120 {
121     int size = 0, insertion = 0, location = 0;
122     printf("Enter the number of elements: ");
123     scanf_s("%d", &size);
124     int* array = (int*)malloc(size * sizeof(int));
125
126     for (int i = 0; i < size; i++)
127     {
128         array[i] = i + 1;
129     }
130
131     printf("Enter the element to be inserted: ");
132     scanf_s("%d", &insertion);
133
134     printf("Enter its location: ");
135     scanf_s("%d", &location);
136
137     array = (int*)realloc(array, (size+1) * sizeof(int)); //to resize the dynamically allocated array
138
139     int counter = size;
140     for (counter; counter >= location; counter--)
141     {
142         array[counter] = array[counter - 1];
143     }
144     array[counter] = insertion;
145
146     for (int i = 0; i <= size; i++)
147     {
148         printf("%d ", array[i]);
149     }
150
151     free(array);
152
153
154
```



```
149
150 }
151
152 free(array);
153
154
155 //Ex5
156 void Ex5_Arrays()
157 {
158     int size = 0, search = 0;
159     printf("Enter the number of elements: ");
160     scanf_s("%d", &size);
161     int* array = (int*)malloc(size * sizeof(int));
162     for (int i = 0; i < size; i++)
163     {
164         array[i] = (i + 1)*11;
165     }
166
167     printf("Enter the element to be searched: ");
168     scanf_s("%d", &search);
169
170     for (int i = 0; i < size; i++)
171     {
172         if (array[i] == search)
173         {
174             printf("Number found at location: %d", i + 1);
175             break;
176         }
177     }
178
179     free(array);
180
181
182
183
184 //***** Strings *****
185 //Ex1
186 void Ex1_Strings()
187 {
```

```
179     free(array);
180 }
181
182
183
184
185 //***** Strings *****/
186 //Ex1
187 void Ex1_Strings()
188 {
189     char text[30], c = 0;
190     printf("Enter a string: ");
191     fgets(text, sizeof(text), stdin);
192     printf("Enter a character to find its frequency: ");
193     scanf_s("%c", &c, 1);
194
195     int i = 0, counter = 0;
196     while (text[i] != 0)
197     {
198         if (text[i] == c)
199             counter++;
200         i++;
201     }
202
203     printf("Frequency of %c is %d\n", c, counter);
204 }
205
206
207 //Ex2
208 void Ex2_Strings()
209 {
210     int len = 0, i=0;
211     char text[30];
212     printf("Enter a string: ");
213     fgets(text, sizeof(text), stdin);
214
215     while (text[i] != 0)
216     {
217         len++;
218     }
```

```
206
207 //Ex2
208 void Ex2_Strings()
209 {
210     int len = 0, i=0;
211     char text[30];
212     printf("Enter a string: ");
213     fgets(text, sizeof(text), stdin);
214
215     while (text[i] != 0)
216     {
217         len++;
218         i++;
219     }
220     len--; //because fgets() adds a newline character '\n' to the string before the NULL character
221     printf("Length of your String is: %d", len);
222 }
223
224
225 //Ex3
226 void Ex3_Strings()
227 {
228     char text[30], newText[30] = {0};
229     printf("Enter a string: ");
230     fgets(text, sizeof(text), stdin); //take care that fgets adds the \n character to the string, so its length includes extra 1
231
232     for (int i = 0, j = strlen(text)-2; i < strlen(text)-1; i++, j--)
233     {
234         newText[i] = text[j];
235     }
236
237     printf("%s", newText);
238 }
239
240
241 //***** Main *****/
242 int main()
243 {
244 }
```



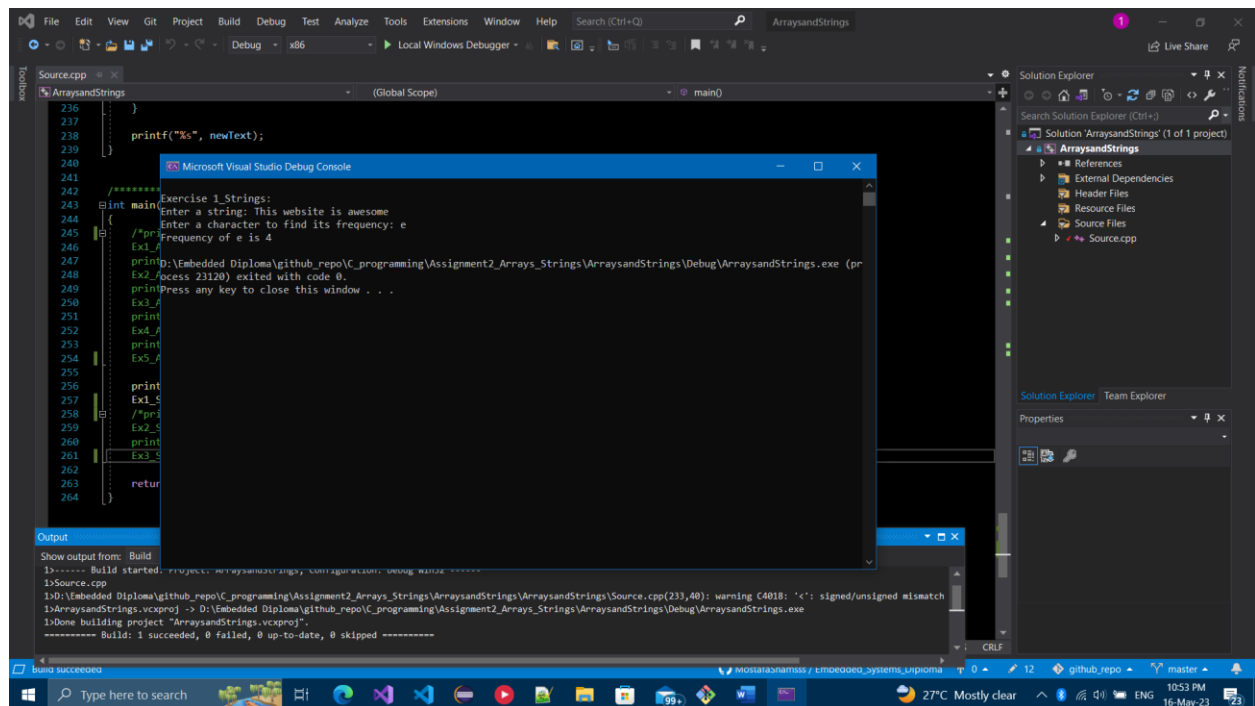
```
D:\Embedded Diploma\github_repo\C_programming\Assignment2_Arrays_Strings\ArraysandStrings\Debug\ArraysandStrings.exe
2.2 0.5
-0.9 25.0

Exercise 2 Arrays:
Enter the numbers of data:
6
Enter number:
45.3
Enter number:
67.5
Enter number:
-45.6
Enter number:
20.34
Enter number:
33
Enter number:
45.6
Average = 27.69

Exercise 3 Arrays:
Enter the rows and columns of a matrix:2
3
Enter elements of a matrix:
Enter element a11: 1
Enter element a12: 2
Enter element a13: 9
Enter element a21: 0
Enter element a22: 4
Enter element a23: 7
Entered matrix:
1 2 9
0 4 7
Transpose of a Matrix:
1 0
2 4
9 7

Exercise 4 Arrays:
Enter the number of elements: 5
Enter the element to be inserted: 6
Enter its location: 2
1 6 2 3 4 5

Exercise 5 Arrays:
Enter the number of elements: 5
Enter the element to be searched: 44
Number found at location: 4
```



```
Microsoft Visual Studio Debug Console

Exercise 2.Strings:
Enter a string: Programiz
Length of your String is: 9

Exercise 3.Strings:
Enter a string: Pritesh
hsetirP
D:\Embedded Diploma\github_repo\C_programming\Assignment2_Arrays_Strings\ArraysandStrings\Debug\ArraysandStrings.exe (process 28252) exited with code 0.
Press any key to close this window . . .
```