```c
 1  #include <stdio.h>
 2  #include <ctype.h>
 3  #include <string.h>
 4
 5  #define MAX_STRING_LENGTH 1024
 6
 7  struct CharacterCount
 8  {
 9      char ch;
10      int ch_count;
11  } character_and_count[] = { { 'A', 0 }, //character_and_count[0].ch = 'A'   &      ↵
       character_and_count[0].ch_count = 0
12                             { 'B', 0 }, //character_and_count[1].ch = 'B'   &      ↵
                       character_and_count[1].ch_count = 0
13                             { 'C', 0 }, //character_and_count[2].ch = 'C'   &      ↵
                       character_and_count[2].ch_count = 0
14                             { 'D', 0 }, //character_and_count[3].ch = 'D'   &      ↵
                       character_and_count[3].ch_count = 0
15                             { 'E', 0 }, //character_and_count[4].ch = 'E'   &      ↵
                       character_and_count[4].ch_count = 0
16                             { 'F', 0 },
17                             { 'G', 0 },
18                             { 'H', 0 },
19                             { 'I', 0 },
20                             { 'J', 0 },
21                             { 'K', 0 },
22                             { 'L', 0 },
23                             { 'M', 0 },
24                             { 'N', 0 },
25                             { 'O', 0 },
26                             { 'P', 0 },
27                             { 'Q', 0 },
28                             { 'R', 0 },
29                             { 'S', 0 },
30                             { 'T', 0 },
31                             { 'U', 0 },
32                             { 'V', 0 },
33                             { 'W', 0 },
34                             { 'X', 0 },
35                             { 'Y', 0 },
36                             { 'Z', 0 } }; //character_and_count[25].ch = 'Z'   &   ↵
                       character_and_count[25].ch_count = 0
37
38  #define SIZE_OF_ENTIRE_ARRAY_OF_STRUCTS sizeof(character_and_count)
39  #define SIZE_OF_ONE_STRUCT_FROM_THE_ARRAY_OF_STRUCTS sizeof(character_and_count   ↵
       [0])
40  #define NUM_ELEMENTS_IN_ARRAY (SIZE_OF_ENTIRE_ARRAY_OF_STRUCTS /                   ↵
       SIZE_OF_ONE_STRUCT_FROM_THE_ARRAY_OF_STRUCTS)
41
42  // ENTRY POINT FUNCTION
43  int main(void)
44  {
```

```c
45        //variable declarations
46        char str[MAX_STRING_LENGTH];
47        int i, j, actual_string_length = 0;
48
49        //code
50        printf("\n\n");
51        printf("Enter A String : \n\n");
52        gets_s(str, MAX_STRING_LENGTH);
53
54        actual_string_length = strlen(str);
55
56        printf("\n\n");
57        printf("The String You Have Entered Is : \n\n");
58        printf("%s\n\n", str);
59
60        for (i = 0; i < actual_string_length; i++)
61        {
62            for (j = 0; j < NUM_ELEMENTS_IN_ARRAY; j++) //Run every character of the
                  input string through the entire alphabet (A TO Z)
63            {
64                str[i] = toupper(str[i]); //If input character is in lower case, turn
                      it to upper case for comparison
65
66                if (str[i] == character_and_count[j].ch) //If character is present...
67                    character_and_count[j].ch_count++; //Increment its count by 1 ...
68            }
69        }
70
71    printf("\n\n");
72    printf("The Number Of Occurences Of ALL Characters From The Alphabet Are As
          Follows : \n\n");
73    for (i = 0; i < NUM_ELEMENTS_IN_ARRAY; i++)
74    {
75        printf("Character %c  =  %d\n", character_and_count[i].ch,
              character_and_count[i].ch_count);
76    }
77    printf("\n\n");
78
79    return(0);
80 }
81
```