```c
 1  #include <stdio.h>
 2
 3  #define MAX_STRING_LENGTH 512
 4
 5  int main(void)
 6  {
 7      //variable declaraions
 8
 9      // *** A 'STRING' IS AN ARRAY OF CHARACTERS ... so char[] IS A char ARRAY AND
10         HENCE, char[] IS A 'STRING' ***
10      // *** AN ARRAY OF char ARRAYS IS AN ARRAY OF STRINGS !!! ***
11      // *** HENCE, char[] IS ONE char ARRAY AND HENCE, IS ONE STRING ***
12      // *** HENCE, char[][] IS AN ARRAY OF char ARRAYS AND HENCE, IS AN ARRAY OF
         STRINGS ***
13
14      //Here, the string array can allow a maximum number of 5 strings (5 rows) and
            each of these 5 strings can have only upto 10 characters maximum (10
         columns)
15      char strArray[5][10]; // 5 ROWS (0, 1, 2, 3, 4) -> 5 STRINGS (EACH STRING CAN
         HAVE A MAXIMUM OF 10 CHARACTERS)
16      int char_size;
17      int strArray_size;
18      int strArray_num_elements, strArray_num_rows, strArray_num_columns;
19      int i;
20
21      //code
22      printf("\n\n");
23
24      char_size = sizeof(char);
25
26      strArray_size = sizeof(strArray);
27      printf("Size Of Two Dimensional ( 2D ) Character Array (String Array) Is = %d
         \n\n", strArray_size);
28
29      strArray_num_rows = strArray_size / sizeof(strArray[0]);
30      printf("Number of Rows (Strings) In Two Dimensional ( 2D ) Character Array
         (String Array) Is = %d\n\n", strArray_num_rows);
31
32      strArray_num_columns = sizeof(strArray[0]) / char_size;
33      printf("Number of Columns In Two Dimensional ( 2D ) Character Array (String
         Array) Is = %d\n\n", strArray_num_columns);
34
35      strArray_num_elements = strArray_num_rows * strArray_num_columns;
36      printf("Maximum Number of Elements (Characters) In Two Dimensional ( 2D )
         Character Array (String Array) Is = %d\n\n", strArray_num_elements);
37
38      // *** PIECE-MEAL ASSIGNMENT ***
39      // ****** ROW 1 / STRING 1 ******
40      strArray[0][0] = 'M';
41      strArray[0][1] = 'y';
42      strArray[0][2] = '\0'; //NULL-TERMINATING CHARACTER
43
```

```c
44        // ****** ROW 2 / STRING 2 ******
45        strArray[1][0] = 'N';
46        strArray[1][1] = 'a';
47        strArray[1][2] = 'm';
48        strArray[1][3] = 'e';
49        strArray[1][4] = '\0'; //NULL-TERMINATING CHARACTER
50
51        // ****** ROW 3 / STRING 3 ******
52        strArray[2][0] = 'I';
53        strArray[2][1] = 's';
54        strArray[2][2] = '\0'; //NULL-TERMINATING CHARACTER
55
56        // ****** ROW 4 / STRING 4 ******
57        strArray[3][0] = 'P';
58        strArray[3][1] = 'r';
59        strArray[3][2] = 'a';
60        strArray[3][3] = 'd';
61        strArray[3][4] = 'n';
62        strArray[3][5] = 'y';
63        strArray[3][6] = 'a';
64        strArray[3][7] = '\0'; //NULL-TERMINATING CHARACTER
65
66        // ****** ROW 5 / STRING 5 ******
67        strArray[4][0] = 'G';
68        strArray[4][1] = 'o';
69        strArray[4][2] = 'k';
70        strArray[4][3] = 'h';
71        strArray[4][4] = 'a';
72        strArray[4][5] = 'l';
73        strArray[4][6] = 'e';
74        strArray[4][7] = '\0'; //NULL-TERMINATING CHARACTER
75
76        printf("\n\n");
77        printf("The Strings In the 2D Character Array Are : \n\n");
78
79        for (i = 0; i < strArray_num_rows; i++)
80            printf("%s ", strArray[i]);
81
82        printf("\n\n");
83
84        return(0);
85  }
86
87  int MyStrlen(char str[])
88  {
89        //variable declarations
90        int j;
91        int string_length = 0;
92
93        //code
94        // *** DETERMINING EXACT LENGTH OF THE STRING, BY DETECTING THE FIRST
              OCCURENCE OF NULL-TERMINATING CHARACTER ( \0 ) ***
```

```
 95         for (j = 0; j < MAX_STRING_LENGTH; j++)
 96         {
 97             if (str[j] == '\0')
 98                 break;
 99             else
100                 string_length++;
101         }
102         return(string_length);
103 }
104
```