```
#include <stdio.h>
 2 #include <stdlib.h>
 3
 4 #define MAX STRING LENGTH 512
 5
 6 int main(void)
 7 {
 8
        //function prototype
 9
        void MyStrcpy(char *, char *);
10
        int MyStrlen(char *);
11
12
        //variable declarations
        char *chArray_Original = NULL, *chArray_Copy = NULL; // A Character Array >>
13
          Is A String
14
        int original_string_length;
15
16
       //code
17
18
       // *** STRING INPUT ***
19
        printf("\n\n");
        chArray_Original = (char *)malloc(MAX_STRING_LENGTH * sizeof(char));
20
21
        if (chArray_Original == NULL)
22
23
           printf("MEMORY ALLOCATION FOR ORIGINAL STRING FAILED !!! EXITTING
              NOW...\n\n");
24
           exit(0);
25
        }
26
27
        printf("Enter A String : \n\n");
        gets_s(chArray_Original, MAX_STRING_LENGTH);
28
29
30
        original_string_length = MyStrlen(chArray_Original);
31
        chArray_Copy = (char *)malloc(original_string_length * sizeof(char));
32
        if (chArray Copy == NULL)
33
        {
           printf("MEMORY ALLOCATION FOR COPIED STRING FAILED !!! EXITTING NOW... →
34
              n\n";
35
           exit(0);
36
        }
37
        // *** STRING COPY ***
38
39
        MyStrcpy(chArray_Copy, chArray_Original);
40
        // *** STRING OUTPUT ***
41
        printf("\n\n");
42
        printf("The Original String Entered By You (i.e : 'chArray_Original') Is : >
43
           n\n";
44
        printf("%s\n", chArray_Original);
45
46
        printf("\n\n");
        printf("The Copied String (i.e : 'chArray_Copy') Is : \n\n");
47
        printf("%s\n", chArray_Copy);
48
49
50
        if (chArray_Copy)
51
        {
52
           free(chArray_Copy);
```

```
... a rameter \verb|\05-StringOperations| \verb|\02-StringCopy| StringCopy.c|
```

chArray Copy = NULL;

53

```
P
```

```
54
             printf("\n\n");
             printf("MEMORY ALLOCATED FOR COPIED STRING HAS BEEN SUCCESSFULLY
55
               FREED !!!\n\n");
56
         }
57
         if (chArray_Original)
58
59
60
             free(chArray_Original);
61
             chArray_Original = NULL;
             printf("\n\n");
62
             printf("MEMORY ALLOCATED FOR ORIGINAL STRING HAS BEEN SUCCESSFULLY
63
               FREED !!!\n\n");
64
         }
65
66
         return(0);
67 }
68
69 void MyStrcpy(char *str_destination, char *str_source)
70 {
71
         //function prototype
72
         int MyStrlen(char *);
73
74
         //variable declarations
75
         int iStringLength = 0;
         int j;
76
77
        //code
78
79
         iStringLength = MyStrlen(str_source);
         for (j = 0; j < iStringLength; j++)</pre>
80
81
             *(str destination + j) = *(str source + j);
82
         *(str_destination + j) = '\0';
83
84 }
85
86 int MyStrlen(char *str)
87
88
         //variable declarations
89
         int j;
90
         int string_length = 0;
91
92
        //code
         // *** DETERMINING EXACT LENGTH OF THE STRING, BY DETECTING THE FIRST
93
           OCCURENCE OF NULL-TERMINATING CHARACTER ( \0 ) ***
         for (j = 0; j < MAX STRING LENGTH; j++)</pre>
94
95
         {
96
             if (str[j] == '\0')
97
                 break;
98
             else
99
                 string_length++;
100
101
         return(string length);
102
    }
103
```