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

```
\underline{\dots} arameter \verb|\|05-Str| ingOperations \verb|\|02-Str| ingCopy \verb|\|Str| ingCopy \verb|\|.c|
```

```
2
```

```
53
             chArray_Copy = NULL;
             printf("\n\n");
 54
             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
    void MyStrcpy(char *str_destination, char *str_source)
 69
 70 {
 71
         //function prototype
         int MyStrlen(char *);
 72
 73
 74
         //variable declarations
 75
         int iStringLength = 0;
         int j;
 76
 77
 78
        //code
 79
         iStringLength = MyStrlen(str source);
 80
         for (j = 0; j < iStringLength; j++)</pre>
 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 ) ***
 94
         for (j = 0; j < MAX_STRING_LENGTH; j++)</pre>
 95
96
             if (str[j] == '\0')
 97
                 break;
 98
             else
 99
                 string_length++;
100
101
         return(string_length);
102 }
103
```