```c
#include <stdio.h>
#include <stdlib.h>

#define MAX_STRING_LENGTH 512

int main(void)
{
    //function prototype
    void MyStrcpy(char *, char *);
    int MyStrlen(char *);

    //variable declarations
    char *chArray_Original = NULL, *chArray_Copy = NULL; // A Character Array Is A String
    int original_string_length;

    //code

    // *** STRING INPUT ***
    printf("\n\n");
    chArray_Original = (char *)malloc(MAX_STRING_LENGTH * sizeof(char));
    if (chArray_Original == NULL)
    {
        printf("MEMORY ALLOCATION FOR ORIGINAL STRING FAILED !!! EXITTING NOW...\n\n");
        exit(0);
    }

    printf("Enter A String : \n\n");
    gets_s(chArray_Original, MAX_STRING_LENGTH);

    original_string_length = MyStrlen(chArray_Original);
    chArray_Copy = (char *)malloc(original_string_length * sizeof(char));
    if (chArray_Copy == NULL)
    {
        printf("MEMORY ALLOCATION FOR COPIED STRING FAILED !!! EXITTING NOW... \n\n");
        exit(0);
    }

    // *** STRING COPY ***
    MyStrcpy(chArray_Copy, chArray_Original);

    // *** STRING OUTPUT ***
    printf("\n\n");
    printf("The Original String Entered By You (i.e : 'chArray_Original') Is : \n\n");
    printf("%s\n", chArray_Original);

    printf("\n\n");
    printf("The Copied String (i.e : 'chArray_Copy') Is : \n\n");
    printf("%s\n", chArray_Copy);

    if (chArray_Copy)
    {
        free(chArray_Copy);
```

```
53          chArray_Copy = NULL;
54          printf("\n\n");
55          printf("MEMORY ALLOCATED FOR COPIED STRING HAS BEEN SUCCESSFULLY
              FREED !!!\n\n");
56      }
57
58      if (chArray_Original)
59      {
60          free(chArray_Original);
61          chArray_Original = NULL;
62          printf("\n\n");
63          printf("MEMORY ALLOCATED FOR ORIGINAL STRING HAS BEEN SUCCESSFULLY
              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;
76      int j;
77
78      //code
79      iStringLength = MyStrlen(str_source);
80      for (j = 0; j < iStringLength; j++)
81          *(str_destination + j) = *(str_source + j);
82
83      *(str_destination + j) = '\0';
84  }
85
86  int MyStrlen(char *str)
87  {
88      //variable declarations
89      int j;
90      int string_length = 0;
91
92      //code
93      // *** DETERMINING EXACT LENGTH OF THE STRING, BY DETECTING THE FIRST
          OCCURENCE OF NULL-TERMINATING CHARACTER ( \0 ) ***
94      for (j = 0; j < MAX_STRING_LENGTH; j++)
95      {
96          if (str[j] == '\0')
97              break;
98          else
99              string_length++;
100     }
101     return(string_length);
102 }
103
```