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

#define MAX_STRING_LENGTH 512

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

    //variable declarations
    char *chArray = NULL; //Character Array Can Be Represented By A char pointer to Mark The Base Address (char *)
    int iStringLength = 0;

    //code
    printf("\n\n");
    chArray = (char *)malloc(MAX_STRING_LENGTH * sizeof(char));
    if (chArray == NULL)
    {
        printf("MEMORY ALOCATION TO CHARACTER ARRAY FAILED !!! EXITTING NOW... \n\n");
        exit(0);
    }

    // *** STRING INPUT ***
    printf("Enter A String : \n\n");
    gets_s(chArray, MAX_STRING_LENGTH);

    // *** STRING OUTPUT ***
    printf("\n\n");
    printf("String Entered By You Is : \n\n");
    printf("%s\n", chArray);

    // *** STRING LENGTH ***
    printf("\n\n");
    iStringLength = MyStrlen(chArray);
    printf("Length Of String Is = %d Characters !!!\n\n", iStringLength);

    if (chArray)
    {
        free(chArray);
        chArray = NULL;
    }

    return(0);
}

int MyStrlen(char *str)
{
    //variable declarations
    int j;
    int string_length = 0;

    //code
    // *** DETERMINING EXACT LENGTH OF THE STRING, BY DETECTING THE FIRST
```

```
          OCCURENCE OF NULL-TERMINATING CHARACTER ( \0 ) ***
55      for (j = 0; j < MAX_STRING_LENGTH; j++)
56      {
57          if (*(str + j) == '\0')
58              break;
59          else
60              string_length++;
61      }
62      return(string_length);
63  }
64
```