```c
#include <stdio.h>

#define MAX_STRING_LENGTH 512

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

    //variable declarations
    char *chArray_Original = NULL, *chArray_Reversed = 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);

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

    MyStrrev(chArray_Reversed, 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 Reversed String (i.e : 'chArray_Reversed[]') Is : \n\n");
    printf("%s\n", chArray_Reversed);

    if (chArray_Reversed)
    {
        free(chArray_Reversed);
        chArray_Reversed = NULL;
        printf("\n\n");
        printf("MEMORY ALLOCATED TO REVERSED STRING HAS BEEN SUCCESSFULLY FREED !!!\n\n");
    }

    if (chArray_Original)
    {
        free(chArray_Original);
        chArray_Original = NULL;
        printf("\n\n");
        printf("MEMORY ALLOCATED TO ORIGINAL STRING HAS BEEN SUCCESSFULLY FREED !!!\n\n");
    }

    return(0);
}

void MyStrrev(char *str_destination, char *str_source)
{
    //function prototype
    int MyStrlen(char *);

    //variable declarations
    int iStringLength = 0;
    int i, j, len;

    //code
    iStringLength = MyStrlen(str_source);

    // ARRAY INDICES BEGIN FROM 0, HENCE, LAST INDEX WILL ALWAYS BE (LENGTH - 1)
    len = iStringLength - 1;

    // WE NEED TO PUT THE CHARACTER WHICH IS AT LAST INDEX OF 'str_source' TO THE FIRST INDEX OF 'str_destination'
    // AND SECOND-LAST CHARACTER OF 'str_source' TO THE SECOND CHARACTER OF 'str_destination' and so on...
    for (i = 0, j = len; i < iStringLength, j >= 0; i++, j--)
    {
        *(str_destination + i) = *(str_source + j);
    }

    *(str_destination + i) = '\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 ) ***
    for (j = 0; j < MAX_STRING_LENGTH; j++)
    {
        if (str[j] == '\0')
            break;
        else
            string_length++;
    }
    return(string_length);
}

```