



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

---

Objetivo: Ejemplificación del uso de punteros y arrays. Recursividad. Arrays de caracteres. Uso de punteros y aritmética de punteros en lugar del uso de índices para arrays unidimensionales. Funciones de conversión y búsqueda. unión, estructura y tipos de datos.

### **Ejemplificación del uso de punteros y arrays.**

Al respecto refiérase a la presentación del primer día de clase ("A modo de introducción"), desde la página 172 en adelante.

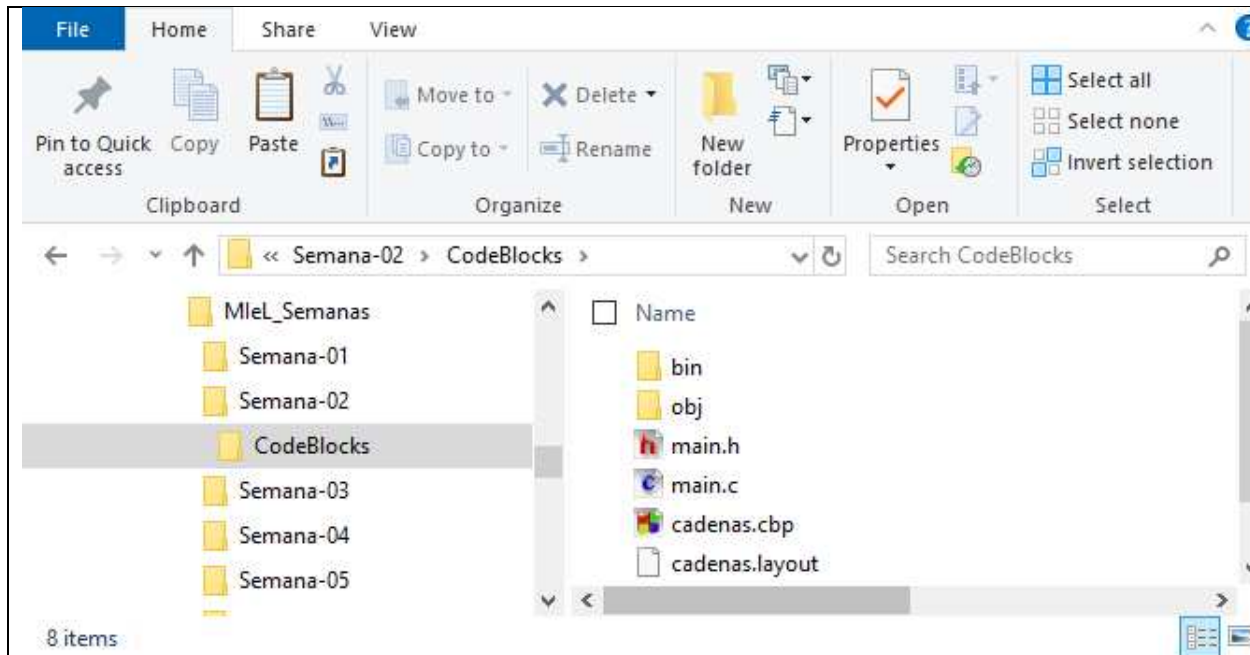
A continuación, le presentaremos algún ejemplo (vea "Funciones de la biblioteca `<string.h>`", también del primer día de clase, -fin de la pág. 5, comienzo de la pág. 6-).

La función `strcpy` de la biblioteca `<string.h>`, recibe en su primer argumento en qué dirección de memoria se quiere copiar un array de enteros char (coloquialmente: dónde quiero copiar una *cadena* de caracteres), y en su segundo argumento desde dónde se quiere copiar. Está sobreentendido que el lugar en que se quiere copiar tiene reservada suficiente memoria para contener la copia (coloquialmente: ¡es un problema *cerebral* del programador si *todo vuela por los aires*! Académicamente: la precondición para copiar es que el destino tenga lugar suficiente, y que el origen sea una cadena que tenga marca de fin de cadena (`'\0'`), caso contrario habrá aleatoriamente resultados erróneos o fallas inesperadas con violaciones de memoria o cambio del contenido de otras variables). Esta función termina de copiar en cuanto encuentra el carácter nulo (`'\0'`), lo copia, y devuelve la dirección de memoria que recibió en su primer argumento.

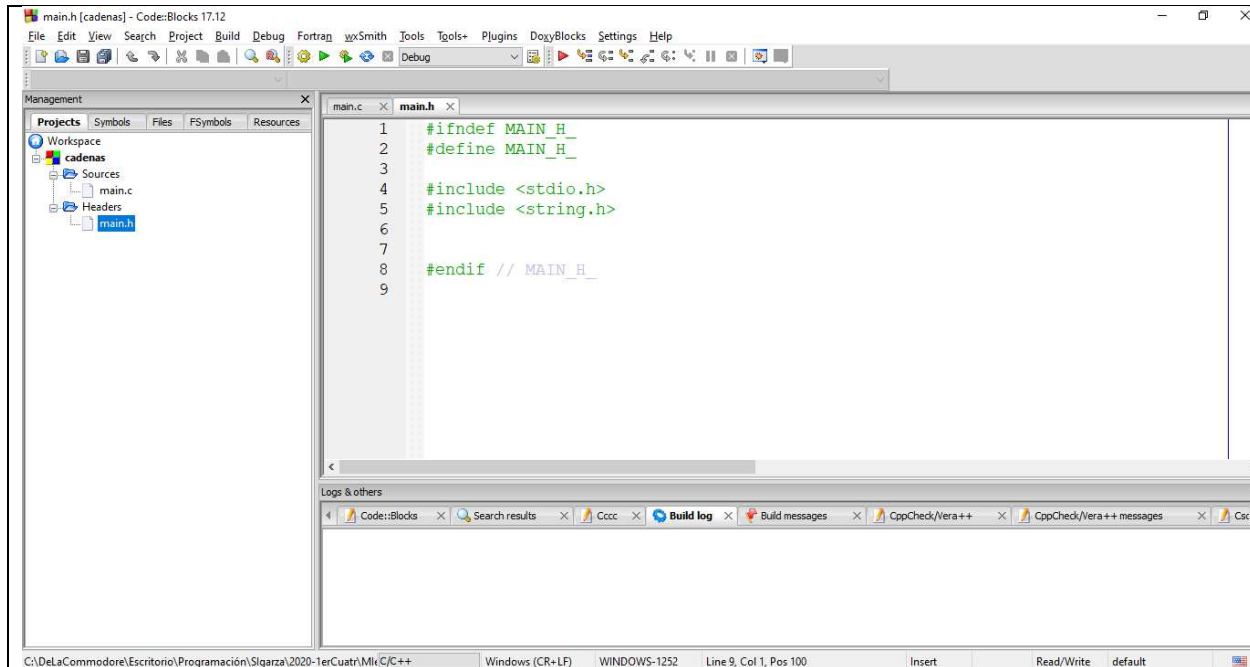
El ejemplo de código ahí escrito es una demostración (tomada del libro de K. y R., de la potencia del Lenguaje C, que permite escribir en muy pocas líneas de código, aunque con una pequeña pérdida de tiempo de ejecución), del código eficiente que resuelve el problema.



Comencemos por generar un proyecto (yo lo he llamado cadenas y lo generé en un subdirectorio `CodeBlocks`), al que le agregué `main.h`, dónde se hacen los `#include` de las bibliotecas utilizadas, por ahora `<string.h>` y `<stdio.h>`.



`main.h` se ve del siguiente modo (disculpen: suelo quitarle el sufijo `INCLUDED` ...



... a las etiquetas de las directivas de compilación condicional **#ifndef** y **#define** cuando genero los ".h" con el entorno).

Y como por ahora mi objetivo es ver cómo se comporta la función de biblioteca **strcpy**, escribo el código correspondiente a tener declarada una cadena de caracteres en la que copiar con suficiente espacio (array de enteros **char: 'destino'**, de veinte posiciones), la cadena a copiar (array de enteros **char: 'origen'**, pre inicializado con una "cadena de caracteres", a la que el compilador le dará cinco posiciones, una por cada '**carácter**' más una para el carácter nulo), además de una variable puntero a char, **aux**, para recibir lo que devuelve la invocación a la función **strcpy**.



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```

1
2  #include "main.h"
3
4  int main()
5  {
6      char    destino[20],
7             original[] = { "Hola" },
8             *aux;
9
10     aux = strcpy(destino, original);
11     printf("La cadena original es:    \"%s\"\n"
12           "la cadena destino queda con \"%s\"\n"
13           "strcpy devolvio %p y la cadena destino esta en la direccion %p\n",
14           original, destino, aux, destino);
15     return 0;
16 }
17

```

Logs & others

```

----- Run: Debug in cadenas (compiler: GNU GCC Compiler) -----
Checking for existence: C:\DeLaCommodore\Escritorio\Programación\Sigarza\2020-1erCuatr\MiE_L_Semanas\Semana-02\CodeBlocks\bin\Debug\cadenas.exe
Executing: "C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe" "C:\DeLaCommodore\Escritorio\Programación\Sigarza\2020-1erCuatr\MiE_L_Semanas\Semana-02\CodeBlocks\bin\Debug\cadenas.exe" (in C:\DeLaCommodore\Escritorio\Programación\Sigarza\2020-1erCuatr\MiE_L_Semanas\Semana-02\CodeBlocks\.)

```

```

C:\DeLaCommodore\Escritorio\Programación\Sigarza\2020-1erCuatr\MiE_L_Semanas\Semana-02\CodeBlocks\bin\Debug\cadenas.exe
La cadena original es:    "Hola"
la cadena destino queda con "Hola"
strcpy devolvio 0060FEE8 y la cadena destino esta en la direccion 0060FEE8

Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.

```

NOTE que para mostrar las cadenas se utilizó la secuencia de escape ('\\'') antes y después de ("%s"). Para mostrar los punteros se usó ("%p").

¿Sabe que la cadena de control de impresión de la función `printf` puede cortarse y continuarse sin que deje de ser una sola cadena de control de impresión como se ve en el código escrito y que esto se hace para no sobrepasar las 80 columnas?

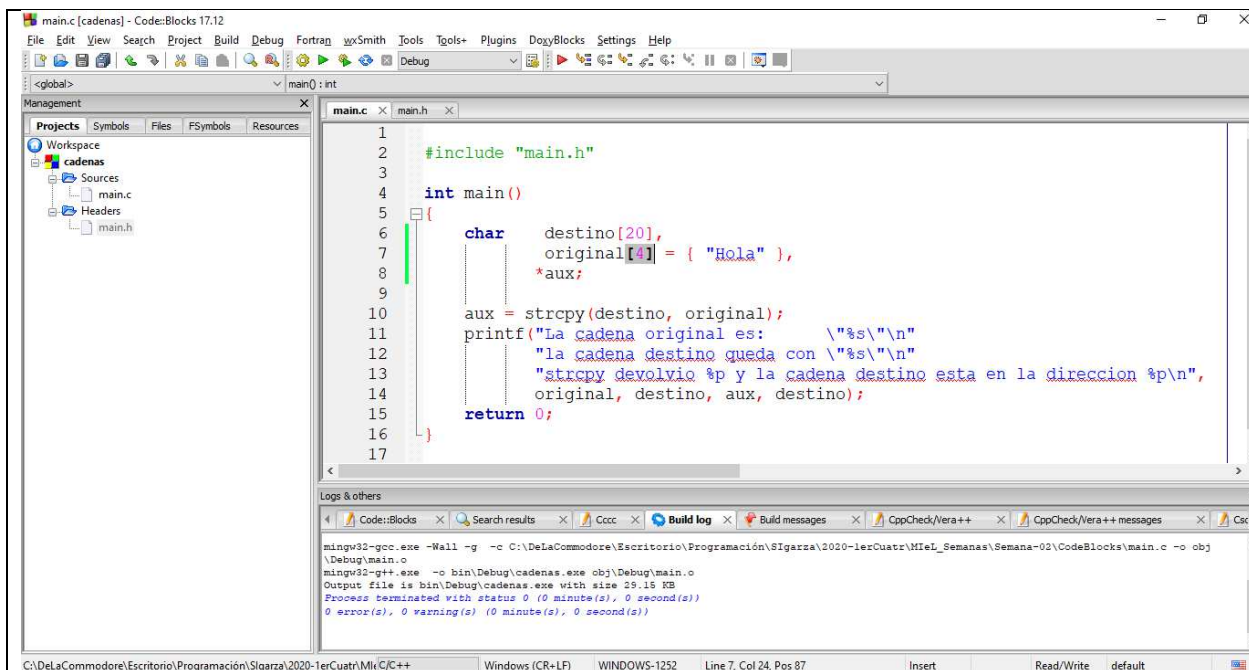
Esto es una pauta de estilo, con la que debe tener escritos sus programas.

¿Le queda claro que con ("%p") se puede la dirección de memoria de un puntero?

¿Entiende que el identificador con que se declara un array tiene implícita la dirección de memoria del primer elemento del mismo?

Bien, esto es válido para arrays de elementos de cualquier tipo.

¿Qué pasaría si el array origen estuviera declarado con exactamente cuatro posiciones?,  
¿compila?



```
1 #include "main.h"
2
3
4 int main()
5 {
6     char destino[20],
7         original[4] = { "Hola" },
8         *aux;
9
10    aux = strcpy(destino, original);
11    printf("La cadena original es:   \"%s\\n\"
12          "la cadena destino queda con \"%s\\n\"
13          "strcpy devolvio %p y la cadena destino esta en la direccyn %p\\n",
14          original, destino, aux, destino);
15    return 0;
16 }
17
```

Logs & others

```
mingw32-gcc.exe -Wall -g -o C:\DeLaCommodore\Escritorio\Programación\Sigarza\2020-1erCuatr\MiEj_Semanas\Semana-02\CodeBlocks\main.c -o obj\Debug\main.o
mingw32-g++.exe -o bin\Debug\cadenas.exe obj\Debug\main.o
Output file is bin\Debug\cadenas.exe with size 29.15 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```

¡Si, compila!, pero al ejecutarlo, bajo ciertas condiciones ...



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```
main.c [cadenas] - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DovyBlocks Settings Help
C:\DeLaCommodore\Escritorio\Programación\Sigarza\2020-1erCuatr\MiEl_Semanas\Semana-02\CodeBlocks\bin\Debug\cadenas.exe
Process returned -1073741819 (0xC0000005) execution time : 2.250 s
Press any key to continue.

6 char destino[20],
7 original[4] = { "HOLA" },
8 *aux;
9
10 aux = strcpy(destino, original);
11 printf("La cadena original es: %s\n",
12 "la cadena destino queda con %s\n",
13 "strcpy devolvio %p y la cadena destino esta en la direccion %p\n",
14 original, destino, aux, destino);
15 return 0;
16 }
17

Logs & others
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Csc
----- Run: Debug in cadenas (compiler: GNU GCC Compiler)-----
Checking for existence: C:\DeLaCommodore\Escritorio\Programación\Sigarza\2020-1erCuatr\MiEl_Semanas\Semana-02\CodeBlocks\bin\Debug\cadenas.exe
Executing: "C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe" "C:\DeLaCommodore\Escritorio\Programación\Sigarza\2020-1erCuatr\MiEl_Semanas\Semana-02\CodeBlocks\bin\Debug\cadenas.exe" (in C:\DeLaCommodore\Escritorio\Programación\Sigarza\2020-1erCuatr\MiEl_Semanas\Semana-02\CodeBlocks\.)

C:\DeLaCommodore\Escritorio\Programación\Sigarza\2020-1erCuatr\MiEl_Semanas\Semana-02\CodeBlocks\bin\Debug\cadenas.exe
Windows (CR+LF) WINDOWS-1252 Line 7, Col 24, Pos 87 Insert Read/Write default 04:07 1/4/2020
```

... produce una violación de memoria, en otros casos puede dar resultados inesperados.

Podría, incluso, dar la violación de memoria en una función hecha por nosotros. Eso nos daría una gran ayuda en el momento de depurar nuestra función.

¿Qué pasaría si el array origen estuviera inicializado con bastante más de veinte caracteres?





UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

The screenshot shows the Code::Blocks IDE with a project named 'cadenas'. The main.c file contains the following code:

```

1
2  #include "main.h"
3
4  int main()
5  {
6      char    destino[20],
7             original[] = { "Hola, es la hora de quedarse en casa, ya que"
8                           "estamos en medio de una pandemia, tomando los recaudos del caso" },
9             *aux;
10
11     aux = strcpy(destino, original);
12     printf("La cadena original es:      \"%s\"\n"
13           "la cadena destino queda con \"%s\"\n"
14           "strcpy devolvio %p y la cadena destino esta en la direccion %p\n",
15           original, destino, aux, destino);
16     return 0;
17 }

```

The 'Logs & others' window shows the following output:

```

----- Run: Debug in cadenas (compiler: GNU GCC Compiler) -----
Checking for existence: C:\DeLaCommodore\Escritorio\Programacion\Sigarza\2020-1erCuatr\Miel_Semanas\Semana-02\CodeBlocks\bin\Debug\cadenas.exe
Executing: "C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe" "C:\DeLaCommodore\Escritorio\Programacion\Sigarza\2020-1erCuatr\Miel_Semanas\Semana-02\CodeBlocks\bin\Debug\cadenas.exe" (in C:\DeLaCommodore\Escritorio\Programacion\Sigarza\2020-1erCuatr\Miel_Semanas\Semana-02\CodeBlocks\bin\Debug\cadenas.exe)

```

Compila perfectamente, pero ...

The command prompt window shows the output of the program:

```

La cadena original es:      "Hola, es la hora de quedarse en casa, ya queestamos en medio de una pande
mia, tomando los recaudos del caso"
la cadena destino queda con "Hola, es la hora de 
strcpy devolvio 0060FED8 y la cadena destino esta en la direccion 0060FED8

Process returned -1073741819 (0xC0000005)   execution time : 1.828 s
Press any key to continue.

```

... después de copiar los veinte caracteres, ¡produce una violación de memoria!



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

---

**Uso de punteros y aritmética de punteros en lugar del uso de índices para arrays unidimensionales.**

Hagamos nuestra propia versión de "`strcpy`", de hecho, haremos varias y para evitar el uso de la de biblioteca las llamaremos "`strcpy1`", "`strcpy2`", etc...

Pero en primer lugar analicemos el problema:

Para resolver un problema hay que tener en cuenta varios pasos:

- entender el problema (y esto se debe enfatizar, porque no es trivial)
- buscar una o varias estrategias de solución
- si hay casos en que la estrategia pueda fallar (casos de "contorno" o "borde"), considerar cuáles son y si no se los puede incluir en la solución, especificar cuáles son las precondiciones de la solución, o sea que se excluyan los "casos de borde" que no se pueden controlar
- elegir una de las estrategias y llevarla a la práctica

En nuestro caso, como vamos a resolver nuestra propia versión de una función de biblioteca, el primer punto y el tercero ya están explicitados. Sólo nos falta "entender el problema", "buscando una estrategia de solución razonable", luego podremos intentar explorar otras estrategias.

**Continúe viendo la presentación del próximo ítem en la plataforma (parte B).**

**Siga con la presentación del siguiente ítem en la plataforma (Parte C).**

**Termine viendo la presentación del siguiente ítem en la plataforma (Parte D).**