**Novedades con la libreria CS50.h de Harvard: Matrices o Arrays de cadenas de texto (no confundir con "caracteres").**

Inicialmente hay que decir que, la libreria de CS50.h, no recurre al tipo de dato "char" (propio de caracteres) para crear cadenas de texto, no. No es lo más correcto para dicha libreria.

**CS50** fue más allá y creó un tipo de dato propio para las cadenas de texto (strings); de tal manera que, para declarar una "string" no se hace con "char"; sino, con un tipo de dato llamado "string", casualmente. Es algo totalmente nuevo y facilita las cosas.

*Sintaxis:* ***string var = "escriba cadena de texto aquí";***

***nota:*** *función que también consta de; una función para solicitar como dato de entrada una cadena de texto o “string” propiamente, se llama: "get\_string". Se ve algo más o menos así al ser declarada:*

*Sintaxis:* ***string var = get\_string ("inserte cadena de texto aquí");***

En ese orden de ideas, la libreria de cs50 se adelantó y también creó una función, no sólo para crear propiamente cadenas de texto ni para recibirlas; sino, también, para habilitar la opción de construir, dentro del tipo de dato "string", matrices o Arrays de cadenas textos!

*Tal que así,*

***string var[x];*** *// donde "x" es el número de elementos (o cadenas de texto) que queremos almacenar dentro de la matriz de tipo "string".* Es decir, dentro de una variable podríamos tener almacenadas varias cadenas de texto, las cuales se identificarían por medio de posiciones que van desde el número 0 hasta el número x: [x].

Esto es completamente nuevo. Originalmente, en la librería estándar de C, stdio.h, sólo existen matrices de caracteres: lo que vendría siendo realmente una sola cadena de texto (string) una sola nada más; pero, ahora con esta función nueva, propia de cs50., podríamos también crear una matriz de cadenas de textos (strings) y no sólo de caracteres... *donde podemos agrupar en una misma variable varias cadenas de texto, esto según la posición de las mismas dentro de la matriz de tipo "string".*

Una completa novedad, como hemos comentado. Antes sólo podíamos tener una cadena de texto en nuestra matriz de carácter, ahora podemos tener propiamente una matriz de cadenas de textos que nos permita agrupar varias cadenas (de texto) clasificadas por posiciones.

Entonces, recapitulemos, el equivalente de declarar una matriz de caracteres de forma estándar en C (una cadena de texto), con la libreria **stdio.h**, es: ***char var [x];*** en cambio, en **cs50**, solamente se declara una nueva función llamada: *string,* y se le pasa una variable; tal que así:

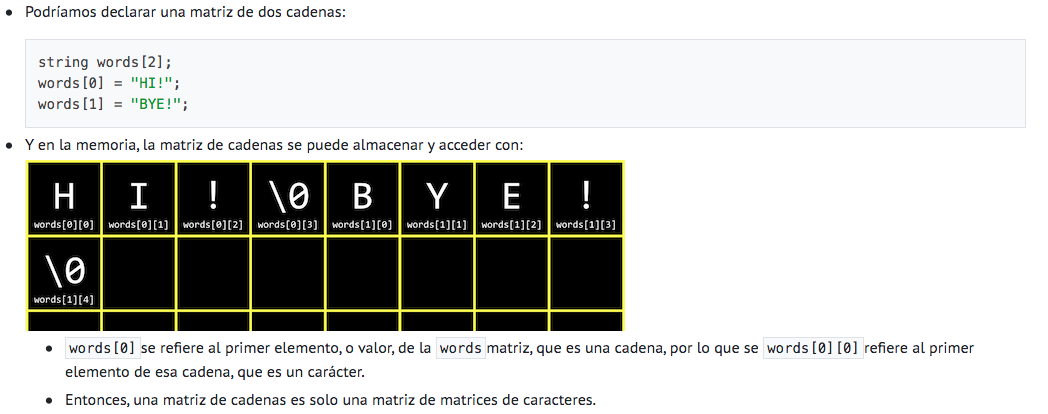
***string var = "cualquier cosa que desee escribir en esta cadena";***...

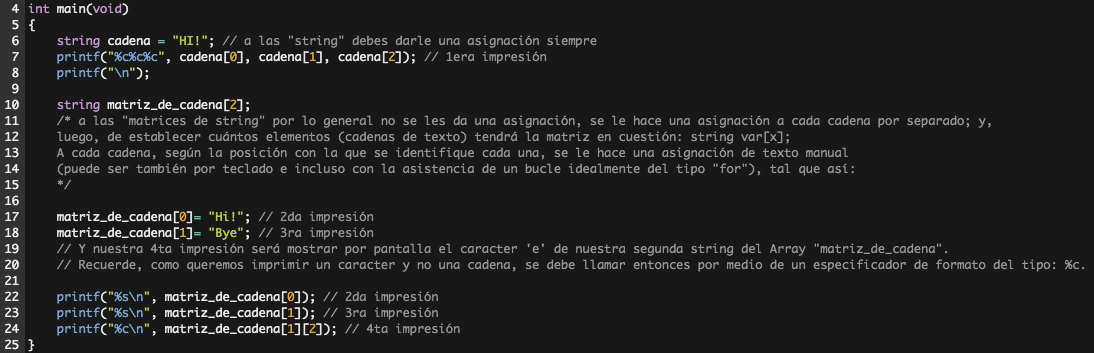
Con esta función puede acceder también a los elementos de la cadena (a sus caracteres); tal que así: si tenemos una string con nombre de “var”, por ejemplo: ***string var = “hola!”;*** *...**entonces, para acceder a sus caracteres, digamos, al primer caracter de la cadena de texto:* ***“hola!”****, sería así:* ***var[0]*** *(claro, asistiendose de un especificador de formato del tipo “%c” -es de tipo “%c” porque realmente estás intentando acceder a un caracter-).* ***Donde var[0] equivale al primer caracter de la cadena de texto “var”, el cual es la letra: “h”.***

Pues bien, como ya se dijo, en cs50 también hay arrays para strings y no sólo para caracteres; tal que así: ***string var [x];*** donde, dentro de las llaves [], se debe especificar el número de elementos (strings) que queremos que tenga nuestro array de Strings.

Y también podría inicializarlas una por una (las cadenas de texto) o solicitarlas una a una por teclado; interesante, no? ***Toda una matriz de... matrices de caracteres (lo que es lo mismo: strings).***

***string var[];  
var[0] = “cadena de texto 1”;  
var[1] = “cadena de texto 2”;***



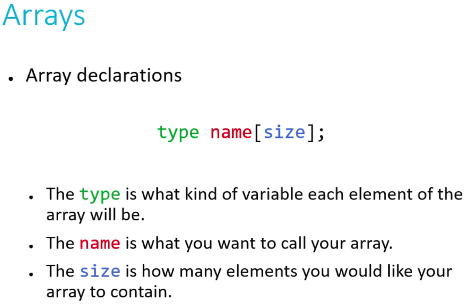


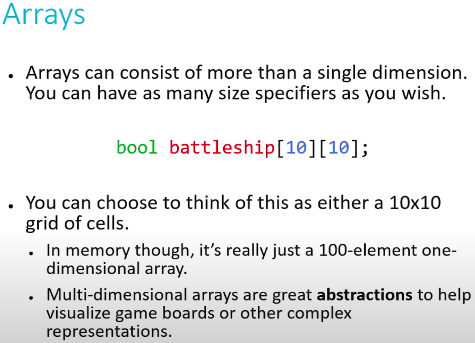
***dato curioso: en la línea 7, si reemplazabamos el especificador de formato %c por uno de tipo entero: %i; pasaría que, entonces, en la salida se imprimiría por pantalla el código ASCII correspondiente a cada caracter de la cadena de texto.***

En las librerías convencionales de C no existe esta posibilidad porque simplemente no existe una función para declarar, propiamente, una cadena. Sólo existe una función, y es lo más cercano, para declarar caracteres (char).

***En resumen... Arrays en términos generales.***

*Las matrices se usan para contener valores del mismo tipo de dato en ubicaciones de memoria contiguas; por lo general, estos valores (llamados elementos) se relacionan entre sí o tienen algo en común que las identifica. Entonces, para evitarnos el engorroso trabajo de darles un nombre propio de variable a cada una y, viendo que, todos esos elementos tienen algo en común por el cual se rigen; en ese caso puntual, idealmente se declara una matriz (Array). Para acceder a cada uno de esos elementos, en el eventual caso de que así lo desee, se hace por medio de lo que se conoce como: número de índice. El primer número de índice, ya se recordará, es: var[0].*





*Ahora, una matriz con area de 10\*10 celdas, siendo el total de ellas 100:*