Uso de arreglos tipo carácter para almacenar y manipular cadenas

- Un arreglo de caracteres se puede inicializar mediante el uso de una literal de cadena. Por ejemplo:

```
char cadena1[] = "primero";
```

- Los arreglos de caracteres también se pueden inicializar mediante constantes tipo carácter individuales en una lista inicializadora.

```
char cadena1[] = { 'f', 'i', 'r', 's', 't','\0'};
```

- Podemos introducir una cadena directamente en un arreglo de caracteres mediante el teclado, por lo que la declaración sería:

```
char cadena2[ 20 ];
```

Uso de arreglos tipo carácter para almacenar y manipular cadenas

Debido a que en la representación interna de una cadena de caracteres es terminada por el símbolo ‘\0', para un texto de "n" caracteres, debemos reservar "n+1”. El carácter ‘\0', aunque pertenece a la cadena, no aparece al utilizar funciones como printf.

Uso de arreglos tipo carácter para almacenar y manipular cadenas

Debido a que en la representación interna de una cadena de caracteres es terminada por el símbolo ‘\0', para un texto de "n" caracteres, debemos reservar "n+1”. El carácter ‘\0', aunque pertenece a la cadena, no aparece al utilizar funciones como printf.

Biblioteca string.h

- **strlen(<cadena>):** Devuelve la longitud de la cadena sin tomar en cuenta el carácter de final de cadena.
- **strcpy(<cadena_destino>, <cadena_origen>):** Copia el contenido de <cadena_origen> en <cadena_destino>.
- **strcat(<cadena_destino>, <cadena_origen>):** Concatena el contenido de <cadena_origen> al final de <cadena_destino>.
- **strcmp(<cadena1>, <cadena2>) :** Compara las dos cadenas y devuelve un 0 si las dos cadenas son iguales, un número negativo si <cadena1> es menor que (precede alfabéticamente a) <cadena2> y un número positivo (mayor que cero) si <cadena1> es mayor que <cadena2>.

Paso de arreglos a funciones

- Para pasar un argumento tipo arreglo a una función, se debe especificar el nombre del arreglo sin corchetes.

```
int temperaturasPorhora[ 24 ];
```

- la llamada a la función sería:

```
modificarArreglo( temperaturasPorHora, 24 );
```

- Para que una función reciba un arreglo a través de la llamada a una función, la lista de parámetros de la función debe especificarlo:

```
void modificarArreglo( int b[], int tamanioArreglo )
```

- Este prototipo se podría haber escrito así:

```
void modificarArreglo( int [], int );
```

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

int main ()
{
    char cadena1[] = "primero segundo tercero";
    printf("Cadena expandida\n");
    for (int i=0; i<strlen(cadena1); i++) {
        printf("%c ",cadena1[i]);
    }
    return 0;
}

```