

Nota: Los siguientes ejercicios se refieren a programas ANSI C no modularizados (autocontenidos dentro de la función `main()`).

Arreglos y cadenas de caracteres (*strings*)

- 0) Explicar qué sucede si se olvida el carácter nulo como último carácter de una cadena (*string*).
- 1) Escribir un programa que lea una cadena de caracteres ingresada por el flujo de entrada estándar y la imprima por pantalla. Para ello utilizar las funciones de biblioteca `scanf()` con formato "%s", `gets()` y `fgets()`. Comparar los resultados y explicar la conveniencia de utilizar cada una de ellas y en qué circunstancias.
- 2) Escribir un fragmento de código que determine si una cadena de caracteres está vacía o no, de dos formas distintas. Utilizar el carácter NUL para una de ellas y la función de biblioteca `strcmp()` para la otra.
- 3) Escribir un fragmento de código que inicialice un arreglo de caracteres con una cadena cualquiera y la imprima por el flujo de salida estándar.
- 4) Escribir un programa que inicialice arreglo de caracteres con una cadena cualquiera y lo imprima en orden inverso por el flujo de salida estándar.
- 5) Escribir un programa que dada una cadena de caracteres, calcule su longitud y la imprima por pantalla (equivalente a la función de biblioteca `strlen()`, encabezado `<string.h>`).
- 6) Escribir un programa que dada una cadena de caracteres y un arreglo de caracteres con espacio suficiente, copie la cadena en el arreglo y coloque el carácter '\0' (equivalente a la función de biblioteca `strcpy()`, encabezado `<string.h>`).
- 7) Escribir un programa que dadas dos cadenas de caracteres realice la concatenación de una sobre la otra y luego termine la cadena con el carácter '\0' (equivalente a la función de biblioteca `strcat()`, encabezado `<string.h>`).
- 8) Escribir un programa que dadas dos cadenas de caracteres, realice una comparación lexicográfica entre ellas y devuelva valores positivos, cero y negativos, según el resultado obtenido (equivalente a la función de biblioteca `strcmp()`, encabezado `<string.h>`).
- 9) Escribir un programa que dadas dos cadenas de caracteres, compare en forma lexicográfica los primeros "n" caracteres y devuelva un valor positivo, cero o negativo, según corresponda (equivalente a la función de biblioteca `strncmp()`, encabezado `<string.h>`).
- 10) Escribir un programa que dada una cadena de caracteres y un arreglo de caracteres con espacio suficiente, copie los primeros "n" caracteres de la cadena sobre el arreglo, sin terminar la cadena con el carácter nulo. (equivalente a la función de biblioteca `strncpy()`, encabezado `<string.h>`).
- 11) Escribir un programa que convierta una cadena de caracteres a mayúsculas (equivalente a la función no ANSI `strlwr()`).
- 12) Escribir un programa que convierta una cadena de caracteres a minúsculas (equivalente a la función no ANSI `strupr()`).
- 13) a) Definir un tipo enumerativo `format_t` compuesto por los símbolos `FMT_MAYUSCULAS` y `FMT_MINUSCULAS`.
b) Escribir un programa que lea del flujo de entrada estándar una cadena de caracteres y la convierta de forma paramétrica a minúsculas o mayúsculas, según una opción ingresada por el usuario mediante `stdin`, y muestre el resultado de la conversión por `stdout`. Usar tipos enumerativos para la decodificación de la selección ingresada por el usuario.
- 14) Escribir un programa que lea de `stdin` una cadena de caracteres, un carácter viejo y un carácter nuevo, 1

y reemplace en la cadena todas las apariciones del carácter `viejo` por el carácter `nuevo`.

- 15) Escribir un programa que lea de `stdin` una cadena de caracteres que comience con espacios en blanco y los elimine desplazando los caracteres útiles hacia la izquierda. (operación *left-trim*).
- 16) Escribir un programa que lea de `stdin` una cadena de caracteres que finalice con espacios en blanco y los elimine desplazando los caracteres útiles hacia la izquierda. (operación *right-trim*).
- 17) Escribir un programa que reciba dos cadenas de caracteres denominadas `s1` y `s2` respectivamente, verifique la existencia de la cadena `s2` como subcadena integrante de la `s1` e imprima el resultado de la búsqueda por `stdout` (ver función de biblioteca `strstr()`, encabezado `<string.h>`).
- 18) Escribir un programa que lea una cadena de `stdin`, determine si es un palíndromo o no e informe el resultado por `stdout`. Considerar el caso de longitud de cadena par e impar.
- 19) Indicar si el siguiente código es correcto o no, y justificar adecuadamente la respuesta:

```
char a[] = "hola";  
char b[] = "mundo";  
  
if(a==b) printf("Son iguales.");  
else printf("Son distintas.");
```

- 20) Escribir un programa que lea dos números de `stdin` en formato de cadena de caracteres, las convierta a números de un tipo determinado, los compare y finalmente muestre por `stdout` los resultados de la comparación. Para ello utilizar las funciones `atoi()` y `atof()` de la biblioteca estándar. ¿Qué se puede concluir sobre su uso? ¿Qué desventajas presenta?
- 21) Escribir un programa que lea de `stdin` un número flotante (`float`) y lo convierta a una cadena de caracteres, en forma inversa a la función `atof()`. ¿Existe la función ANSI de biblioteca `ftoa()`? Sugerencia: utilizar la función de biblioteca `sprintf()`.
- 22) Ídem para un número entero (`int`).
- 23) Escribir un programa que lea un número entero positivo de `stdin` y guarde su valor en base octal sobre una cadena de caracteres, mostrando por `stdout` el resultado de la conversión. Sugerencia: utilizar la función de biblioteca `sprintf()`.
- 24) Ídem para base hexadecimal.
- 25) Ídem para base binaria. ¿Existe en lenguaje C un formato de impresión en base binaria?
- 26) Escribir un programa que lea dos cadenas de caracteres y un número. El programa debe insertar la segunda cadena en la primera, a partir del carácter de la primera cadena que está en la posición indicada por el número.
- 27) Escribir un programa que reciba una cadena de caracteres y un número entero no negativo `N` y la ajuste al margen derecho, dentro de un espacio de `N` caracteres.

Arreglos unidimensionales

- 28) Escribir una programa que inicialice un vector de un tipo determinado (`float`, `double` o `int`) e imprima cada uno de sus elementos por `stdout`.

- 29) Escribir un programa que permita ingresar n números enteros en un arreglo llamado `numeros` y despliegue el conjunto de sus elementos, así como la suma de ellos.
- 30) Escribir un programa que permita calcular el promedio aritmético de una serie de valores ingresados por teclado almacenando la totalidad de los valores leídos. A tal efecto, se debe ingresar primero la cantidad esperada de elementos a promediar y luego el lote de datos de a uno por vez.

La media se calcula con la siguiente expresión:

$$\bar{x} = \frac{1}{N} \cdot \sum_{i=1}^N x_i$$

Finalizado el ingreso de datos, mostrar el resultado por `stdout` con 3 decimales. Realizar todas las validaciones que considere necesarias.

Modificar el programa del ejercicio anterior para calcular además el desvío estándar del conjunto de datos leídos. Luego mostrar por `stdout` el resultado con 2 decimales. El desvío estándar de un lote de muestras se calcula de acuerdo con la expresión:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(x_i - \bar{x} \right)^2}$$

- 31) En relación a los dos ejercicios anteriores, demostrar matemáticamente que no es necesario almacenar la totalidad de los datos de un lote para realizar el cálculo de la media aritmética o del desvío estándar. Se puede pensar este problema como la lectura de una secuencia de valores, en la cual se recalcula la media en cada iteración (problema conocido como Actualización de la Media). Lo mismo sucede con el cálculo del desvío estándar (problema conocido como Actualización del Desvío Estándar) y para ello se utiliza la media aritmética actualizada.
- 32) Escribir un programa que dado un vector de un tipo determinado (`float`, `double` o `entero`) y una variable de un tipo enumerativo `extreme_t`, halle el máximo o el mínimo elemento del arreglo en función de la variable de control y lo muestre por pantalla.
- 33) Escribir un programa que genere N números aleatorios en un vector de elementos de tipo `int`. Mostrar su contenido por `stdout`.
- 34) Escribir un programa que dado un vector de números y su longitud, determine si dicho vector está ordenado o no.
- 35) Escribir un programa que realice el muestreo de una forma de onda determinada y almacene las N muestras en un vector de elementos de tipo `double`. La forma de onda responde a la expresión:

$$v(t) = A \cdot \sin(2 \cdot \pi \cdot f \cdot t + \varphi)$$

en donde:

A: Amplitud de pico de la onda senoidal.
f: frecuencia de la onda senoidal, en Hz.
t: variable independiente (tiempo).
 φ : fase inicial en radianes.

Luego mostrar el contenido de las muestras por `stdout` con 4 decimales.

¿Cómo las volcaría en un archivo de texto? (Redireccionar la salida por `stdout` a un archivo). Recuperar los resultados con una planilla de cálculo.

- 36) Polinomio. Escribir un programa que lea los coeficientes de un polinomio, su grado y la variable independiente x , e imprima el valor del polinomio evaluado en ese punto. Sugerencia: utilizar la Regla de Horner para realizar el cálculo de modo eficiente.

37) Polinomio. Escribir un programa que realice un muestreo de una función polinómica de grado G y almacene las N muestras en un vector de `double`. Luego imprimir el contenido de las muestras en forma tabular por `stdout` con 3 decimales.

38) Transcodificación. Se tienen los siguientes juegos de caracteres:

UTF-8: Alfabeto "universal" en el cual los símbolos extendidos del castellano se representan según dos bytes.

ISO-8859-1: Alfabeto latino occidental en el que los símbolos se representan dentro de un byte.

Entidades HTML: Una representación como cadenas ASCII de los símbolos extendidos de las codificaciones ISO-8859.

Sea la siguiente tabla de códigos correspondientes al idioma castellano:

Carácter	UTF-8	ISO-8859-1	Entidad
á	c3 a1	e1	á
é	c3 a9	e9	é
í	c3 ad	ed	í
ó	c3 b3	f3	ó
ú	c3 ba	fa	ú
Á	c3 81	c1	Á
É	c3 89	c9	É
Í	c3 8d	cd	Í
Ó	c3 93	d3	Ó
Ú	c3 9a	da	Ú
ü	c3 bc	fc	ü
Ü	c3 9c	dc	Ü
ñ	c3 b1	f1	ñ
Ñ	c3 91	d1	Ñ
¿	c2 bf	bf	¿
¡	c2 a1	a1	¡

Se pide:

- Escribir un programa que convierta textos en UTF-8 a ISO-8859-1.
- Escribir un programa que convierta textos en ISO-8859-1 a UTF-8.
- Escribir un programa que convierta textos en UTF-8 a entidades HTML.

39) Dar un ejemplo de uso de arreglo de caracteres que no corresponda a una cadena de caracteres.

Arreglos multidimensionales: arreglos de cadenas de caracteres

40) Explicar la siguiente declaración: `char cadd[3][10] = {"dia", "mes", "año"};`

- Traducción. a) Definir un tipo enumerativo `mes_t` con 12 símbolos que representen cada uno de los meses del año.
 b) Declarar un arreglo de cadenas de caracteres con los nombres de los 12 meses del año.
 c) Escribir un fragmento de código que, a partir del contenido de una variable de tipo `mes_t`, imprima por pantalla la descripción del mes.
- Modificar el ejercicio anterior para incluir un diccionario de meses, que permita presentar los nombres de los meses en varios idiomas. La decisión del idioma a utilizar será tomada en tiempo de compilación. Sugerencia: utilizar compilación condicional.

Arreglos multidimensionales: matrices

- Escribir un programa que inicialice una matriz cuadrada de $N \times N$ elementos de tipo `double`, con $N=4$, calcule la traza de la matriz y muestre el resultado por el flujo de salida estándar.
- Ídem ejercicio anterior, pero leyendo de forma interactiva cada uno de los valores de la matriz ingresados mediante el flujo de entrada estándar.

- 45) Escribir un programa que calcule el determinante de una matriz de 2 por 2 y de 3 por 3 números reales.
- 46) Escribir un programa que cargue una matriz de enteros de $N \times M$ y obtenga la transpuesta de dicha matriz, para mostrar luego por `stdout` el contenido de la matriz resultante.
- 47) Escribir un programa que cargue una matriz cuadrada de enteros de $N \times N$ a partir de los datos leídos por `stdin` y determine si es una matriz simétrica o no.
- 48) Escribir un programa que genere dos matrices de dimensiones $N \times M$ y $M \times P$ y realice el producto de ellas, para mostrar a continuación la matriz resultante.
- 49) Escribir un programa que inicialice un arreglo bidimensional de números enteros, y encuentre y despliegue el máximo valor contenido en él.