

6. Argumentos en Línea de Órdenes / Command-Line Arguments (CLA)

Nota: Para los siguientes ejercicios, deben realizarse todas las validaciones correspondientes.

- 1) Escribir un programa que calcule la longitud de una cadena de caracteres pasada como argumento en línea de órdenes, e imprima el resultado por `stdout`. ¿Cómo invocaría al programa para calcular la longitud de la expresión: $2 * 4 = 8$?
- 2) Escribir un programa que compare lexicográficamente dos cadenas de caracteres pasadas como argumentos en línea de órdenes, y muestre el resultado de la comparación por `stdout`.
- 3) Escribir un programa (CLA) que reciba como argumento un número natural, y calcule su factorial, mostrando su resultado por `stdout`.
- 4) Escribir un programa (CLA) reciba como argumentos una serie de cadenas de caracteres, y las muestre ordenadas alfabéticamente por `stdout`.
- 5) Escribir un programa (CLA) que muestre por `stdout` el contenido del directorio actual en el que se encuentra.

Nota: Puede utilizar el programa del ejercicio 4). ¿Cómo realizaría, en ese caso, la invocación. (Ayuda: usar metacaracteres del shell de comandos).

- 6) Escribir un programa (CLA) que permita convertir un texto (cadena de caracteres) escrito en charset ASCII a entidades de ISO-8859-1, devolviendo el resultado por el `stdout`.
- 7)
 - a) Escribir un programa que reciba como argumento la cantidad de números aleatorios a generar y un carácter delimitador, y devuelva por el `stdout` el lote de números aleatorios generados (como strings), delimitados por el carácter separador.
 - b) Redireccione el `stdout` y vuelque los resultados a un archivo de texto.
 - c) Analice mediante una planilla de cálculo, la aleatoriedad de los números contenidos en dicho archivo.
- 8) Escriba un programa que reciba por línea de comandos una cadena de caracteres como primer argumento y cualquiera de las siguientes opciones como segundo:

`/M` MUESTRA LA CADENA DE CARACTERES EN PANTALLA EN MAYÚSCULAS.

`/m` muestra la cadena de caracteres en pantalla en minúsculas.

y procese la cadena de caracteres en consecuencia. Puede utilizar las funciones `int tolower(int c)` y `int toupper(int c)` de la biblioteca estándar para convertir un carácter a minúscula o mayúscula respectivamente.

- 9)
 - a) Escribir un programa que reciba como argumentos:
 - la cantidad de números aleatorios a generar (tamaño del lote).
 - límite inferior del intervalo.
 - límite superior del intervalo.
 - carácter delimitador.y devuelva por el `stdout` un lote de números aleatorios, cuyos valores estén comprendidos dentro del intervalo cuyos límites fueron recibidos como argumentos, y delimitados por el carácter separador.
 - b) Redireccione el `stdout` y vuelque los resultados a un archivo de texto.
 - c) Verifique mediante una planilla de cálculo, la aleatoriedad de los números generados.
- 10) Escribir un programa (CLA) que reciba como argumentos una serie de cadenas de caracteres, y las imprima por `stdout`, pero formateadas con entidades ISO-8859-1. Para tal fin, el programa debe invocar al desarrollado en el ejercicio 6).

- 11)** ¿Cómo modificaría el programa del ejercicio anterior, si se deseara convertir los datos a charset UTF-8? ¿Y a Unicode? Indicar cómo podría parametrizarse la conversión a otros charsets.

(Nota: NO se pide escribir ni implementar el algoritmo de conversión a UTF-8, sino solamente consultar las funciones de la biblioteca `<iconv.h>`).

- 12)** Utilizando los códigos desarrollados en los ejercicios 4), 10) y 11), indicar cómo podría escribir¹ un programa que reciba una serie de cadenas de caracteres, las ordene alfabéticamente, y las muestre por `stdout` convertidas al juego de caracteres ISO-8859-1.

(Sol. Usar piping: `conv xxx | ordenar | formatear yyy`)

¹: Combinando pequeños programas que permitan solucionar en forma práctica el problema.