

# Ejercicios de IC

Propuestos en *Fundamentos de Programación* como repaso y auto-evaluación

28 de enero de 2019

- 1.- Diseñar un programa que devuelva el máximo de cuatro números.
- 2.- Diseñar un programa que tome tres valores de entrada y muestre por pantalla si uno de ellos es igual a la suma de los otros dos.
- 3.- Diseñar un programa que tome tres longitudes de entrada y nos diga si se puede construir un triángulo con varitas de esas longitudes.
- 4.- Diseñar un programa que devuelva el cociente y el resto de una división entera suponiendo que no disponemos de los operadores cociente (//) y resto (%).
- 5.- Diseñar un programa para obtener el máximo común divisor según el algoritmo de Euclides.
- 6.- Diseñar un programa para calcular la potencia de dos números suponiendo que no disponemos del operador de exponenciación (\*\*).
- 7.- Diseñar una función para calcular la cuota mensual que hay que pagar por una hipoteca, de acuerdo a la siguiente fórmula (con tasa de interés mensual y el plazo especificado en número de meses):

$$cuota = \frac{capital * interes}{100 * (1 - (1 + (interes/100))^{-plazo})}$$

- 8.- Diseñar una función que calcule el factorial de un número.
- 9.- Diseñar una función que aproxime las funciones  $e^x$  y  $\sin(x)$  aplicando los siguientes desarrollos en serie de potencias:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$
$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

- 10.- Diseñar una función que calcule el máximo común divisor de una secuencia de números introducidos por el usuario (véase el ejercicio 5).
- 11.- Los polinomios de Hermite tienen la siguiente forma:

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= 2x \\ H_2(x) &= 4x^2 - 2 \\ \dots &= \dots \\ H_n(x) &= 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x) \end{aligned}$$

Diseñar una función que permita calcular el polinomio de Hermite de grado  $n$

12.- Los polinomios de Legendre tienen la siguiente forma:

$$\begin{aligned}L_0(x) &= 1 \\L_1(x) &= x \\2L_2(x) &= 3x^2 - 1 \\&\dots = \dots \\nL_n(x) &= (2n-1)xL_{n-1}(x) - (n-1)L_{n-2}(x)\end{aligned}$$

Diseñar una función que permita calcular el polinomio de Legendre de grado  $n$

- 13.- Diseñar una función para saber si, dadas dos cadenas, una es la inversa de la otra.
- 14.- Escribir una función que compruebe si una cadena **s** es un palíndromo (lo que comúnmente se conoce como capicúa).
- 15.- Escribir un programa que, dada una cadena de caracteres que contiene palabras separadas por un espacio en blanco, muestre por pantalla aquellas palabras que tienen más de cinco caracteres (Nota: se debe utilizar el método *split*).
- 16.- Escribir una función que, tomando como entrada una cadena de caracteres que contiene palabras separadas por un espacio en blanco, devuelva la misma cadena sin los espacios en blanco. (Nota: resolver el problema (a) usando el método *split* y (b) sin usarlo).
- 17.- Escribir una función que, tomando una cadena de texto como entrada, retorne otra cadena con todas sus consonantes al principio y todas sus vocales al final, eliminando los blancos. Por ejemplo, pasándole la cadena: "curso de programacion", una posible solución sería: "rmcsrpdrgcnuoioa".
- 18.- Escribir una función `ajusta_derecha(s,1)` que, tomando como entrada una cadena de caracteres **s** y una longitud **l** (entero, valor por defecto:  $l = 20$ ), devuelva una cadena de longitud **l** con la cadena **s** ajustada a la derecha y rellenada con espacios en blanco por la izquierda.
- 19.- Dada una cadena en la que cada vez que aparece la palabra **number** a continuación aparecen un espacio en blanco y un número real, escribir una función que calcule la suma de todos los números reales que aparezcan en la cadena.
- 20.- Diseñar una función que cuente el número de caracteres de un archivo de texto.
- 21.- Diseñar una función que cuente el número de apariciones de una determinada palabra en un archivo de texto.
- 22.- Diseñar una función que imprima el porcentaje de palabras de un archivo de texto que no tienen la letra **e**.
- 23.- Diseñar una función que retorne un booleano indicando si una determinada palabra **w** aparece en un archivo de texto **f**.
- 24.- Diseñar una función que muestre por pantalla la línea mas larga de un archivo de texto **f**.
- 25.- Diseñar una función que, tomando como entrada un archivo de texto, cree un nuevo archivo de texto con el contenido del primero salvo las líneas en blanco, que son borradas.
- 26.- Escribir una función que tome como argumentos un archivo de texto **f** y una palabra **w** y cree dos nuevos archivos **g** y **h**: el primero con todas las líneas de **f** que contengan la palabra **w**, y el segundo con todas las líneas de **f** que no la contengan.
- 27.- Escribir una función que tome como argumento un archivo de texto **f**. En cada línea del archivo habrá un número entero (puede ser positivo o negativo). La función leerá todos los números y devolverá el máximo.
- 28.- Diseñar una función que construya una lista con los  $n$  primeros números pares.

- 29.- Diseñar una función que, dada una cadena de entrada  $s$ , genere dos listas: una con todas las palabras de  $s$  en minúsculas y otra con todas las palabras de  $s$  en mayúsculas.
- 30.- Diseñar una función que, dada una lista de enteros, sustituya cada uno de los números pares por una cadena de caracteres que lo represente.
- 31.- Diseñar una función que elimine todos los elementos duplicados de una lista (dejando en ella una sola aparición de cada elemento).
- 32.- Diseñar una función que busque un elemento en una lista ordenada (sin usar ningún método asociado a la lista) y devuelva la posición donde lo encuentre o `None` si no lo encuentra.
- 33.- Diseñar una función que permita mezclar dos secuencias ordenadas de enteros  $a$  y  $b$  en una única secuencia  $c$ , como sigue:
  - a)  $c = a \cup b$  (sin duplicados)
  - b)  $c = a + b$  (con duplicados)
- 34.- Diseñar una función que ordene los elementos de una lista siguiendo el método de la burbuja.
- 35.- Diseñar una función que indique si una matriz es simétrica o no.
- 36.- Diseñar una función que haga la suma o la resta de dos matrices, en función de un parámetro booleano de entrada.
- 37.- Diseñar una función que calcule el histograma de palabras diferentes que hay en un archivo de texto.
- 38.- Diseñar una función que tome como entrada un histograma de palabras y un entero  $n$  y devuelva la lista de palabras que aparecen exactamente  $n$  veces.
- 39.- Diseñar una función que utilice un diccionario para representar el histograma inverso de palabras de un archivo de texto: cada clave será un entero  $n$ , al que le corresponde la lista de palabras que aparecen exactamente  $n$  veces en el archivo.
- 40.- Considerese un archivo de texto con información sobre las asignaturas en que se han matriculado los alumnos de la Facultad. En cada línea del archivo, el texto que aparece a la izquierda del carácter ':' contiene el nombre y apellidos del alumno/a, mientras que a la derecha se enumeran las asignaturas en las que está matriculado/a, separadas por comas. Por ejemplo:

Arantza Torres Asensio: Algebra, Química, Introducción a la Computación  
Lander González de Tena Ugarte: Introducción a la Computación, Física

Diseñar una función que procese un archivo como el citado y cree un diccionario con el número de alumnos matriculados en cada asignatura.