COLEGIO LA CRUZ
SEMPER ALTIUS
Departamento de Matemática

EJERCICIOS – LISTAS PARTE 1

Pensamiento computacional y programación

---

0. Escribe un programa que pida números enteros al usuario y los guarde en una lista. Tu programa debe pedir números hasta que el usuario ingrese 0 (debes indicárselo al usuario). Luego deberás mostrar todos los valores ingresados por el usuario (a excepción del 0) en orden de menor a mayor. Utiliza el método sort o sorted para ordenar la lista.

1. Escribe un programa que pida números enteros al usuario y los guarde en una lista. Usa el 0 como valor centinela para finalizar los inputs. Luego deberás mostrar todos los valores (a excepción del 0) en orden inverso al que fueron ingresados.

2. Al analizar datos recolectados en experimentos científicos es muy común remover aquellos valores extremos antes de realizar cálculos. Escribe una función que reciba dos parámetros: una lista de valores y un número entero **n** no negativo. Tu función deberá retornar una nueva lista donde se han removido los **n** valores más grandes y los **n** valores más pequeños. El orden de los valores que se entreguen en la lista no tiene que coincidir con aquella recibida por la función. No es necesario que imprimas esta lista en pantalla (parte del ejercicio 3).

3. Escribe un programa que ponga en uso la función anterior. Este programa deberá pedir una lista de datos al usuario, un número entero no negativo y deberá imprimir en pantalla la lista de datos resultantes. Además, tu programa debe mostrar un mensaje de error si el usuario ingresa un número entero menor que la cantidad de datos que contiene la lista. Investiga como recibir una lista de datos como input del usuario.

4. En este ejercicio crearás un programa que reciba palabras del usuario hasta que se ingrese un string vacío (presionar enter). Luego, tu programa deberá imprimir en pantalla cada palabra ingresada solo una vez y en orden de aparición. Por ejemplo:

Input del usuario	Output del programa (Print)
primera segunda primera tercera segunda tercera tercera	primera segunda tercera

5. Escribe un programa que reciba números enteros hasta que se ingrese una línea vacía (enter). Tu programa deberá mostrar en pantalla todos los números negativos, seguidos de todos los 0 y luego todos los números positivos. Dentro de cada grupo, los números deberán mostrarse en el mismo orden en que fueron ingresados. Por ejemplo, si se ingresan los valores 3, -4, 1, 0 , -1, 0 y -2, entonces su programa deberá imprimir -4, -1, -2, 0, 0, 3

6. Un número entero positivo **n** es divisor de otro, **m**, si existe un número entero **p** tal que **n**·**p**=**m**, en otras palabras, si al dividir **m** entre **n**, su resultado es exacto. Escribe una función que reciba un número positivo como parámetro y retorne una lista con todos los divisores del número. No es necesario imprimir el resultado en pantalla (parte del ejercicio 7).

7. Escribe un programa que pida al usuario un número entero e imprima en pantalla la lista de todos sus divisores. Para completar este ejercicio deberás importar el programa anterior y utilizar la función que ya programaste (recuerda que, para importar un programa externo, este debe encontrarse en la misma carpeta que tu programa).

8. Un número entero **n** positivo se dice perfecto si la suma de todos sus divisores es igual a **n**. Por ejemplo, el 28 es un número perfecto, por que sus divisores son 1, 2, 4, 7 y 14, y 1 + 2 + 4 + 7 + 14 = 28. Escriba una función que reciba un número entero y retorne True si este es perfecto y False en caso contrario.

9. Escriba un programa que imprima en pantalla una lista que contenga todos los números perfectos entre 1 y 10000.

* Cada uno de los ejercicios de esta guía que realices de forma satisfactoria y cumpliendo todos los requerimientos propuestos, otorga 1pp. Si tu código presenta una solución novedosa/original o destaca por incluir funcionalidades que no se piden, puedes obtener como máximo 2pp.