

IES SAN JUAN DE LA RAMBLA
DPTO. INFORMÁTICA Y COMUNICACIONES
CFGS DESARROLLO DE APLICACIONES MULTIPLATAFORMA
PROGRAMACIÓN 1º

San Juan de la Rambla a 22 de noviembre de 2022

Unidad 03 - Estructuras de datos — Prueba de desarrollo

Esta actividad es la parte de desarrollo de la prueba de esta unidad. Su valor es de 8 puntos. Recuerden que la parte de conocimientos vale 2 puntos.

El peso de esta actividad en el total de la unidad es del 60%.

Esta actividad consiste en la dar solución a tres problemas que se deben resolver mediante un algoritmo escrito en Python.

La forma de entrega es mediante 3 ficheros con extensión .py y con nombre "ejercicio0n.py", cambiando la "n" por el número 1, 2 o 3 según sea el caso

La entrega se realiza a través de la plataforma campus. La prueba comenzará a las 10:45 y terminará a las 13:10 que se cerrará la plataforma.

Mucha suerte!!!!

Ejercicio 01 (4,0 puntos)

Una ferretería que almacena los datos de sus facturas mensuales en un diccionario con la siguiente estructura.

Facturas= {(cod_cliente, nro_factura) : [(cod_articulo, ctdad, preciounitario),...]... }
clave valor

Donde la clave del diccionario es una **tupla** formada por dos campos

(cod_cliente, nro_factura)

Y el valor es una **lista de tuplas** donde cada tupla es como la siguiente:

(cod_articulo, ctdad, precio_unitario)

Se les deja un menú de 4 opciones para contestar cada de las preguntas que se relacionan a continuación. Para obtener las respuestas deberá generar código Python para obtenerlas.

Pregunta 1 (1,0 pts.)

Códigos de los clientes (puede ser 1 o más) que han comprado más (en cantidad de dinero) en la tienda durante este mes.

Pregunta 2 (1,0 pts.)

Códigos de los artículos (puede ser 1 o más) que más se haya vendido (tomando en cuenta la cantidad de artículos).

Pregunta 3 (0,5 pts.)

Número total de facturas realizadas en el mes.

Pregunta 4 (1,5 pts.)

Si se tiene un diccionario Cliente, como el siguiente:

Cliente = { cod cliente : (cif, nombre, dirección), ... }

Indique todos los datos de los clientes que no han realizado compras este mes. No se permite que se imprima la tupla, sino recorrerla para mostrar sus datos.

Sugerencia:

Una forma de atacar algunas de las preguntas puede ser crear una nueva estructura de datos basada en el diccionario dado, donde se pueda obtener de forma más fácil las repuestas solicitadas a los problemas planteados.

Ejercicio 02 (2,0 puntos)

Dada L una lista de lista de números como la siguiente:

`L = [[3, 2, 1], [4, 2, 0], [2, 1, 2], [2, 1, 1], [0, 6, 0]]`

Escribe un programa en Python que permita encontrar la/s lista/s dentro de una lista donde la suma de sus elementos sea la de mayor valor.

En ejemplo anterior, la respuesta que se debe mostrar es:

La mayor suma es 6 y la listas que lo cumplen son:

`[3, 2, 1] [4, 2, 0] [0, 6, 0]`

Otro ejemplo

`L = [[1, 2, 3], [4, 5, 6], [10, 11, 12], [7, 8, 9]]`

En este ejemplo, la respuesta que se debe mostrar es:

La mayor suma es 33 y la listas que lo cumplen son:

`[10, 11, 12]`

Ejercicio 03 (2,0 puntos)

Escribir un programa en Python que permita empaquetar una lista. Empaquetar significa crear una lista de tuplas, donde cada tupla indica la repetición de valores consecutivos de la siguiente forma (elemento_de_la_lista, nro_de_veces_consecutivas_que_aparece). El programa debe imprimir la lista generada. La impresión de la lista debe realizarse recorriendo la misma y en cada fila se debe tener como máximo 5 tuplas. NO SE PUEDE HACER print(lista).

Por ejemplo, empaquetar la lista

[1, 1, 1, 3, 5, 1, 1, 3, 3, 8, 8, 8, 4, 4, 4, 4, 4, 1, 1, 1, 1, 1, 1, 1]

debe generar la siguiente lista de tuplas:

[(1,3) , (3,1) , (5,1) , (1,2) , (3,2), (8,3), (4,5), (1,7)].

El programa debe escribir lo siguiente:

La lista contiene las siguientes tuplas

(1,3) - (3,1) - (5,1) - (1,2) - (3,2)

(8,3) - (4,5) - (1,7)