

Ejercicios - Diccionarios

1. Escribir una función que reciba por parámetro un diccionario con el siguiente formato: { id_producto: [stock_minimo, stock_actual],.....}, donde el id_producto será la clave, de tipo cadena; y la lista asociada a cada clave id_producto, contendrá una dupla de valores, siendo el primero, el stock mínimo a mantener de dicho producto; y el segundo, el stock actual del producto; ambos de tipo entero positivo.
La función debe imprimir un listado con los productos a reponer (cuyo stock_actual sea menor al stock_minimo), indicando el id_producto y la cantidad a reponer.
2. Escribir una función que reciba por parámetro un texto, y devuelva un diccionario, el cual tendrá como claves, cada una de las palabras que hay en el texto, y como valor, la cantidad de ocurrencias de dicha palabra en el texto.
No distinguir entre mayúsculas y minúsculas.
Considerar que las palabras del texto estarán separadas por blancos.
3. Dado el diccionario "dic_materias" que tiene cargados los nombre de las materias como clave, y como valor asociado, una lista con tres números enteros, que indican: la cantidad de alumnos anotados (como primer valor), cantidad de alumnos que rindieron el parcial (segundo valor), cantidad de alumnos que aprobaron el parcial (tercer valor).
Se pide que escribas:
 - a) Una función que reciba el diccionario y devuelva una lista con las materias cuyo índice de deserción sea mayor al 50% (esto se calcula teniendo en cuenta la cantidad de alumnos que rindieron el parcial sobre la cantidad de anotados).
 - b) Una función que reciba el diccionario y que devuelva una lista de tuplas, formadas por pares (materia, porcentaje_aprobados), ordenada de mayor a menor por porcentaje de aprobados (en este caso, se calcula sobre la cantidad que rindieron).
4. Escribir un programa que solicite el ingreso de un texto y luego informe un ranking de las palabras que aparecen en el texto.
El texto ingresado debe tener como mínimo 20 palabras. En caso de no tener las palabras suficientes, solicitar se ingrese más texto, que debe ser agregado al ya existente.
Considere palabras que sólo estén formadas por letras.
Tenga en cuenta que seguido a una palabra, pueden estar los siguientes signos de puntuación: ",;.:", en este caso, se debe quitar el signo.
No debe diferenciar entre mayúsculas y minúsculas, por ejemplo: si aparecen en el texto las siguientes palabras: "sol", "Sol", "SOL"; se deben contabilizar 3 ocurrencias de la palabra "sol" como clave.
No utilice métodos tales como count, find, index.
Imprima el ranking, ordenado descendientemente, por la cantidad de ocurrencias de la palabra.

5. Escribir un programa que permita gestionar los datos de los alumnos de un curso. El programa deberá:
- a)** Permitir la carga de un diccionario, que tendrá por clave un número de padrón, y por valores asociados a la clave, una lista compuesta por valores numéricos, que serán las notas obtenidas por un alumno. En cada ingreso, se deberá solicitar el padrón y la nota a cargar. Si el padrón es existente, se agregará la nota a la lista de notas; sino, se creará la clave padrón correspondiente con la nota asociada. Validar que el padrón sea un número entero entre 1 y 10000; y la nota entre 0 y 10. El ingreso finaliza cuando el padrón ingresado sea igual a cero.
 - b)** Informar mediante un listado, la nota promedio de cada alumno, ordenado por padrón.
 - c)** Informar que alumnos aprobaron la materia y que porcentaje representan. Para aprobar la materia, es necesario que en la lista de notas, al menos, haya 2 notas mayores o iguales a 4. Informar asociado a cada padrón, la nota promedio resultante de sumar sólo los valores mayores ó iguales a 4.
 - d)** En base al punto anterior, informar un ranking de notas, indicando la nota promedio de aprobación, y la cantidad de alumnos que la obtuvieron; ordenado por la cantidad de alumnos.
6. Escribir un programa, compuesto por funciones, que permita:
- a)** Ingresar en un diccionario, localidades (como clave) y dos datos: cantidad de habitantes y cantidad de hospitales públicos (HP). Los datos surgen de distintas planillas, por lo que una misma clave (localidad) se puede ingresar varias veces, debiendo sumarse los valores.
 - b)** Listar el total de habitantes y HP para cada localidad.
 - c)** Imprimir un listado ordenado de mayor a menor de las 5 localidades de mayor relación: (habitantes / HP). Indicar la Localidad y la relación resultante.
7. Escribir un programa que permita administrar los datos de una votación electoral. Para ello, será necesario que nuestro programa cumpla los siguiente requerimientos:
- a)** Gestionar la carga de datos de la votación, almacenando los datos en un diccionario de diccionarios, que tendrá el siguiente formato:
{ provincia: { partido_politico: votos_obtenidos ,,,,} ,,,,}
Por cada una de las provincias, tendremos la cantidad de votos obtenidos por cada uno de los partidos políticos.
La carga de datos debe solicitar, la Provincia, el Partido Político, y la cantidad de votos. Puede haber más de un ingreso de cantidad de votos, para una misma Provincia y Partido Político, en ese caso se deben acumular a los ya existentes.
La carga finaliza cuando se ingrese como nombre de Provincia, la palabra FIN.

- b)** Informar para cada Provincia que Partido obtuvo más votos, indicando la cantidad de votos y el porcentaje que representan sobre el total de las votaciones de la provincia. El listado debe estar ordenado por el nombre de la Provincia.
- c)** Informar el ranking de votos a nivel Nacional por Partido Político. El listado debe estar ordenado de mayor a menor por la cantidad de votos obtenidos, y debe figurar el Partido Político, la cantidad de votos obtenidos, y el porcentaje respecto del total de votos.