
Primer Parcial de Programación 1

05/10/2020

Tener en cuenta que:

- La duración del parcial es de 3 horas.
- Se debe realizar en computadora, para este fin se deberá seguir los siguientes pasos:
 1. Crear una carpeta dentro de su computadora de nombre `primerparcial2020`.
 - En caso de realizar el parcial en una computadora del laboratorio usar la carpeta Z.
 2. Descargar del Moodle el archivo **`primer_parcial_2020.py`** que se encuentra dentro de la carpeta "Parciales Anteriores" dentro de "Primer Parcial"
 3. Mover archivo a carpeta "`primerparcial2020`" creado en paso 1.
 4. Abrir el Visual Studio Code y colocar en el menú la opción `File->Open` ir a la carpeta creada `primerparcial2020` y poner `Open`. Esto debería abrir la carpeta y mostrarle el modulo `primer_parcial_2020.py` a la izquierda para seleccionar.
 5. Dentro de Visual Studio Code seleccionar en el menú la opción `Terminal->New terminal`
 6. En la terminal abierta (abajo) colocar el siguiente comando para verificar que todo esta correcto: `python3 primer_parcial_2020.py`. El resultado esperado es:
 - Prueba 1 incorrecta
 - Prueba 2 incorrecta
 - Prueba 3 incorrecta
 - Prueba 4 incorrecta
- Culinado las horas del parcial se deberá entregar a través del Moodle en la tarea "Entrega Primer Parcial". Se espera que se suba el archivo "**`primer_parcial_2020.py`**" donde coloco el código fuente de la solución.
- Tenga en cuenta que no se corrige código con errores del interprete de Python.
- No se puede usar material, ni uso de internet, ni celulares durante la realización del parcial.
- Se contestan dudas solo en la primer hora y media del parcial.

Ejercicio 1 - Practico (Tiempo recomendado 3:00h)

Una empresa referente del mercado en el rubro de electrodomésticos de cocina quiere pegar un salto en las tecnologías que usa, con el fin de mejorar la experiencia de sus usuarios. Para ello se propuso realizar una primera experiencia, usando uno de sus productos más emblemáticos, la heladera. Uno de los puntos que consideran más relevante atacar con tecnología, es conocer que hay dentro de la heladera sin tener que abrirla. Es muy común que un usuario desee cocinar, y no tenga total certeza de los productos dentro de la heladera, con lo cual tenga que inspeccionar la misma desperdiciando tiempo y energía.

Teniendo esto en mente, se desea realizar un prototipo que valide los algoritmos que van a ser necesarios realizar para poder llevar este proyecto en marcha. Para ello se plantean las siguientes estructuras que van a ser usadas por los distintos algoritmos:

- **heladera**, la heladera se va a representar como un arreglo de dos dimensiones (matriz) que va a indicar los alimentos que contiene. Las dimensiones de la matriz van a ser $N \times 2$, donde N va a ser la cantidad de alimentos en la heladera (filas) y 2 van a indicar por cada alimento su nombre y cantidad en disponible del alimento (columnas). Los líquidos se almacenan en ml y los solidos en gramos o unidades. Un ejemplo de una heladera que contiene 2000ml Leche, 400g Manteca, 400g Tomates y 12 huevos sería el siguiente:
 - `heladera = [["Leche", 2000], ["Manteca", 400], ["Tomate", 400], ["Huevos", 12]]`
- **receta**, el usuario dispone de recetas que es una lista de alimento que debería ser usado para cocinar. En particular de una receta nos interesa aquellos alimentos que se almacenan en frio. Para representar una receta, se va a utilizar una estructura similar a la heladera, donde se tiene una matriz de $M \times 2$, con M es la cantidad de alimentos a utilizar y se tiene dos columnas que indican el nombre del alimento y la cantidad usada (en ml, gramos o

unidades dependiendo del alimento). Un ejemplo de receta de torta yogurt sería el siguiente:

○ `receta_torta = [["Leche", 400], ["Yogurt", 200], ["Huevos", 4]]`

- **tabla_conversion_unidad**, dado que cuando vamos a comprar alimentos muchos de ellos se representan en unidades, es necesario disponer la asociación entre un producto y cuantas cantidades representa una unidad. Por ejemplo, la leche viene en unidades de 1000ml, la manteca viene en unidades de 200ml, con estos dos un ejemplo sería:
 - `tabla_conversion_unidad = [["Leche", 1000], ["Manteca", 200]]`

Con esto en mente, se entrega un módulo de nombre `primer_parcial_2020.py` el cual contiene las siguientes operaciones a implementar:

- **def poner_alimento_heladera(heladera, alimento, cantidad)**
 - dado una estructura de tipo heladera, se agrega a la misma el alimento en las cantidades indicadas. Puede asumir que la cantidad dado el tipo de alimento siempre va a ser ingresado de forma correcta. Por ejemplo, si la leche se mide en ml, es correcto asumir que siempre el valor cantidad va a estar representando en ml para este alimento.
 - Tenga en cuenta que no debería generarse mas de una fila de un mismo alimento.
- **def sacar_alimento_heladera(heladera, alimento, cantidad)**
 - dado una estructura de tipo heladera, se debe sacar el alimento en las cantidades indicadas. En caso de que el alimento no exista o no se tenga suficiente cantidad, se debe retornar `False` y en caso de que se pueda realizar la extracción de forma correcta se debe retornar `True`.
- **def sacar_alimentos_para_receta(heladera, receta)**
 - Esta operación dada una receta debe sacar los elementos de la heladera disminuyendo su stock. Se debe tener en cuenta que los productos se tienen que **poder sacar todos o no se saca ninguno**. Se espera que se retorne como resultado el valor `True` si fue posible extraer todos los productos de la heladera, `Falso` en caso de que no haya sido posible dado que faltaba stock de algún artículo.
- **def alimentos_a_comprar_para_receta(heladera, receta, tabla_conversion_unidad)**
 - Esta operación debe indicar el conjunto de productos a comprar que no se tiene suficiente stock en la heladera para realizar la receta. El resultado esperado es una matriz de $N \times 2$ que tiene una fila por cada alimento que se necesita comprar y cada fila dispone de dos columnas, la primera para indicar el nombre del alimento y la segunda para indicar la cantidad de unidades a comprar. Es importante usar la `tabla_conversion_unidad` para transformar cantidades en unidades, por ejemplo, puede ser necesario para una receta 400 ml de leche, que no se disponga leche en la heladera y sea necesario comprar 1 Leche. Ejemplo:
 - `heladera = [["Huevos", 12]]`
 - `receta = [["Leche", 400], ["Huevos", 3]]`
 - `tabla_conversion_unidad = [["Leche", 1000]]`
 - **resultado esperado** = `[["Leche", 1]]`
 - Puede asumir que si un producto no tiene dato en `tabla_conversion_unidad` el mismo esta representando en unidades (ejemplo Huevo).