75.40 Algoritmos y Programación I Ing. Pablo Guarna Cuat: 02 / 2015 16/02/2016
Padrón: …..…… Nombre y Apellido: ……..……...

La empresa *Mudanzas El Neutral* ofrece un servicio extra a sus clientes para asegurarles que todas las cosas que salen del origen llegan efectivamente a destino. Para esto, los clientes imprimen etiquetas con códigos QR que se pegan a los distintos elementos transportados. Estas etiquetas son escaneadas al salir de origen y al llegar a destino. Se tienen luego dos archivos con el mismo formato:

salida.csv, llegada.csv (ordenados por id creciente):

```
id,hora,volumen_en_m3,peso_en_kg,descripción
```

Se requiere procesar ambos archivos al finalizar la mudanza para:

- Detectar si hubo elementos que no llegaron a destino. En caso de haberlos, se deberá generar un archivo faltantes.txt donde por cada línea se registre: “El artículo descripción con hora de salida hh:mm:ss no llegó a destino”.
- Detectar si hubo pérdida de elementos dentro de los cajones (diferencia de peso en destino inferior en 0,25 kg o más). Registrar en faltantes.txt: “El cajón descripción salió con peso_en_kg kg de origen y llegó con peso_en_kg kg a destino”.
- Informar por pantalla el volúmen y peso total de salida y llegada.
- El formato de la hora en los archivos es hhmmss (por ejemplo, 184802).
- Suponer que los archivos existen y que todas las líneas tienen el formato esperado.
- Los archivos no pueden cargarse en memoria ya que no hay un límite predefinido.
- Considerar que nunca se registran en destino elementos que no salieron en origen.

75.40 Algoritmos y Programación I Ing. Pablo Guarna Cuat: 02 / 2015 16/02/2016
Padrón: …..…… Nombre y Apellido: ……..……...

La empresa *Mudanzas El Neutral* ofrece un servicio extra a sus clientes para asegurarles que todas las cosas que salen del origen llegan efectivamente a destino. Para esto, los clientes imprimen etiquetas con códigos QR que se pegan a los distintos elementos transportados. Estas etiquetas son escaneadas al salir de origen y al llegar a destino. Se tienen luego dos archivos con el mismo formato:

salida.csv, llegada.csv (ordenados por id creciente):

```
id,hora,volumen_en_m3,peso_en_kg,descripción
```

Se requiere procesar ambos archivos al finalizar la mudanza para:

- Detectar si hubo elementos que no llegaron a destino. En caso de haberlos, se deberá generar un archivo faltantes.txt donde por cada línea se registre: “El artículo descripción con hora de salida hh:mm:ss no llegó a destino”.
- Detectar si hubo pérdida de elementos dentro de los cajones (diferencia de peso en destino inferior en 0,25 kg o más). Registrar en faltantes.txt: “El cajón descripción salió con peso_en_kg kg de origen y llegó con peso_en_kg kg a destino”.
- Informar por pantalla el volúmen y peso total de salida y llegada.
- El formato de la hora en los archivos es hhmmss (por ejemplo, 184802).
- Suponer que los archivos existen y que todas las líneas tienen el formato esperado.
- Los archivos no pueden cargarse en memoria ya que no hay un límite predefinido.
- Considerar que nunca se registran en destino elementos que no salieron en origen.

Posible solución

```
#!/usr/bin/python

def leerDeLinea(linea):
   return linea.rstrip('\n').split(',')

def esValida(linea):
   return linea != ''

def procesar_mudanza(salida, llegada, faltantes):

 s_peso_total = 0
 s_vol_total = 0
 l_peso_total = 0
 l_vol_total = 0

 with open(faltantes, "w") as faltantes:
     with open(salida) as archsalida:
         with open(llegada) as archllegada:

             linea = archsalida.readline()
             linea2 = archllegada.readline()


             while esValida(linea) and esValida(linea2):
                 s_id, s_hora, s_peso, s_vol, s_desc = leerDeLinea(linea)
                 l_id, l_hora, l_peso, l_vol, l_desc = leerDeLinea(linea2)

                 if s_id < l_id:
                     faltantes.write("El articulo {} con hora de salida {} no llego a
destino\n"
                                     .format(s_desc, s_hora))

                     s_peso_total += int(s_peso)
                     s_vol_total += int(s_vol)

                     linea = archsalida.readline()

                     #puede resolverse sin el continue, reemplazando el cuerpo del while
por un if..elif..else
                     continue

                 #se supone que s_id == l_id para el resto de los casos por no poder venir
elementos en la llegada
                 #que no esten en la salida
                 if float(s_peso) > float(l_peso) + 0.25:
```

```
                    faltantes.write("El cajon {} salio con {} kg de origen y llego con {}
kg a destino\n"
                                    .format(s_desc, s_peso, l_peso))

                s_peso_total += int(s_peso)
                s_vol_total += int(s_vol)
                l_peso_total += int(l_peso)
                l_vol_total += int(l_vol)

                linea = archsalida.readline()
                linea2 = archllegada.readline()

            while esValida(linea):

                s_id, s_hora, s_peso, s_vol, s_desc = leerDeLinea(linea)

                faltantes.write("El articulo {} con hora de salida {} no llego a
destino\n"
                                    .format(s_desc, s_hora))

                linea = archsalida.readline()

 print("Peso total: {}/{}\nVol. total: {}/{}".format(s_peso_total, l_peso_total,
s_vol_total, l_vol_total))

def main():
   procesar_mudanza('salida.csv', 'llegada.csv', 'faltantes.txt')


if __name__ == '__main__':
   main()
```

Entradas:

salida.csv:

```
1,153400,100,25,caja1
2,153500,200,5,caja2
3,153600,300,75,caja3
4,153700,400,85,caja4
```

llegada.csv:

```
1,153400,100,25,caja1
2,153500,20,50,caja2
3,153600,300,75,caja3
```

Salidas:

consola:

Peso total: 600/420
Vol. total: 105/150

faltantes.txt:

```
El cajon caja2 salio con 200 kg de origen y llego con 20 kg a destino
El articulo caja4 con hora de salida 153700 no llego a destino
```

Entradas:

salida.csv:

```
1,153400,100,25,caja1
2,153500,200,5,caja2
3,153600,300,75,caja3
4,153700,400,85,caja4
```

llegada.csv:

```
1,153400,100,25,caja1
2,153500,200,5,caja2
3,153600,300,75,caja3
4,153700,400,85,caja4
```

Salidas:

consola:

Peso total: 1000/1000
Vol. total: 190/190

faltantes.txt: (vacío)