

Actividades Parte V: Archivos

Sumas filas por columnas

El archivo `datos1.txt` tiene tres números enteros en cada línea:

```
45 12 98
1 12 65
7 15 76
54 23 1
65 2 84
```

1. Escriba la función `suma_lineas(nombre_archivo)` que entregue una lista con las sumas de todas las líneas del archivo:

```
>>> suma_lineas('datos1.txt')
[155, 78, 98, 78, 151]
```

2. Escriba la función `suma_columnas(nombre_archivo)` que entregue una lista con las sumas de las tres columnas del archivo:

```
>>> suma_columnas('datos1.txt')
[172, 64, 324]
```

Reporte de notas

Las notas de un ramo están almacenadas en un archivo llamado `notas.txt`, que contiene lo siguiente:

```
Pepito:5.3:3.7:6.7:6.7:7.1:5.5
Yayita:5.5:5.2:2.0:5.6:6.0:2.0
Fulanita:7.1:6.6:6.4:5.1:5.8:6.3
Moya:5.2:4.7:1.8:3.5:2.7:4.5
```

Cada línea tiene el nombre del alumno y sus seis notas, separadas por dos puntos (":").

Escriba un programa que cree un nuevo archivo llamado `reporte.txt`, en que cada línea indique si el alumno está aprobado (promedio $\geq 4,0$) o reprobado (promedio $< 4,0$):

```
Pepito aprobado
Yayita aprobado
Fulanita aprobado
Moya reprobado
```

Becas a mejores puntuaciones

El Instituto Tecnológico de Putre, por intermedio de su Departamento de Informática, desea llevar un control de todos los puntajes de ingresos obtenidos por los postulantes a dicha universidad.

Fueron mil alumnos los que postularon, y sus puntajes puede encontrarlos [aquí](#)

La cantidad máxima de vacantes es de 850 estudiantes, por lo que deberá seleccionar los mejores puntajes.

También desea premiar a todos los alumnos que ingresen con un puntaje superior a 764 puntos, con una beca mensual de \$60.000.

Con esta información, se pide que desarrolle los siguientes ejercicios.

1. Escriba la función para guardar el contenido del archivo en una lista llamada `obtener_puntajes(archivo)`.
2. Escriba la función para obtener dos listas, una con los alumnos aceptados y otra con los alumnos rechazados, llamada `obtener_listas(alumnos)`
3. Escriba la función llamada `calcular_becas(alumnos)` para poder determinar la cantidad mensual que deberá desembolsar el Instituto debido a las becas que otorgará.
4. Escriba la función llamada `puntaje_promedio(alumnos)` para saber el promedio de los puntajes de las personas aceptadas en la universidad.
5. Utilizando la lista de las personas rechazadas, realice una función que permita obtener la [desviación estándar](#) de sus puntajes.

Becas a mejores puntuaciones

Para los siguientes ejercicios, descargue el archivo [juegos.txt](#). Este archivo tiene una lista de todos los cartones jugados para un sorteo de Loto. Cada línea del archivo tiene la lista de números jugados en un cartón.

Este archivo se puede abrir con cualquier editor de texto para ver su contenido, pero para resolver los problemas, hay que escribir funciones que analicen los datos.

Todas las funciones deben hacer lo siguiente:

- abrir el archivo con `archivo = open('juegos.txt')`
- leer los datos y analizarlos,
- cerrar el archivo con `archivo.close()`

Como cada línea del archivo es un string, hay que convertirlo a un conjunto de números para poder analizarlos, de la siguiente manera:

```
numeros_carton = set()
for n in linea.split():
    numeros_carton.add(int(n))
```

También se puede hacer así:

```
numeros_carton = set(map(int, linea.split()))
```

1. ¿Cuántos cartones de Loto fueron jugados?

(Para responder la pregunta, escriba una función `contar_cartones` que cuente los cartones del archivo).

2. De todos los cartones jugados, ¿cuántos escogieron el número 7?

Para responder la pregunta, escriba una función `contar_numero_en_cartones(n)` que cuente cuántos cartones tienen el número n.

3. Escriba la función `hay_ganadores(numeros)`, cuyo parámetro `numeros` es el conjunto de los seis números de un sorteo, que indique si alguien se ganó el Loto:

```
>>> hay_ganadores({13, 33, 5, 38, 1, 19})
True
>>> hay_ganadores({14, 21, 1, 36, 9, 17})
False
```

4. Escriba la función `n_aciertos(numeros, n)`, que indique cuántas personas tuvieron n aciertos, donde `numeros` es el conjunto de los seis números de un sorteo:

```
>>> n_aciertos({13, 33, 5, 38, 1, 19}, 4)
17
>>> n_aciertos({20, 39, 6, 27, 12, 4}, 3)
229
>>> n_aciertos({1, 2, 3, 4, 5, 6}, 5)
2
```

Cuenta caracteres

Desarrolle un programa que sea capaz de abrir el archivo de texto [de prueba](#) para poder realizar lo siguiente:

Contar cantidad de caracteres en mayúsculas, por ejemplo:

```
>>> mayusculas(texto)
81
```

- Contar cantidad de caracteres en minúsculas, por ejemplo:

```
>>> minusculas(texto)
3481
```

- Contar cantidad de caracteres especiales (ni mayúsculas ni minúsculas):

```
>>> especiales(texto)
136
```

- Determinar el porcentaje de caracteres en minúsculas con respecto a todo el texto, por ejemplo:

```
94.13%
```

- Contar la cantidad de vocales que posee el texto, por ejemplo:

```
>>> vocales(texto)
1524
```

- Eliminar todas las vocales del texto.

Consulta médica

Una consulta médica tiene un archivo `pacientes.txt` con los datos personales de sus pacientes. Cada línea del archivo tiene el rut, el nombre y la edad de un paciente, separados por un símbolo `:`. Así se ve el archivo:

```
12067539-7:Anastasia López:32
15007265-4:Andrés Morales:26
8509454-8:Pablo Muñoz:45
7752666-8:Ignacio Navarro:49
8015253-1:Alejandro Pacheco:51
9217890-0:Patricio Pimienta:39
9487280-4:Ignacio Rosas:42
12393241-2:Ignacio Rubio:33
11426761-9:Romina Pérez:35
15690109-1:Francisco Ruiz:26
6092377-9:Alfonso San Martín:65
9023365-3:Manuel Toledo:38
10985778-5:Jesús Valdés:38
13314970-8:Abel Vázquez:30
7295601-k:Edison Muñoz:60
5106360-0:Andrea Vega:71
8654231-5:Andrés Zambrano:55
10105321-0:Antonio Almarza:31
13087677-3:Jorge Álvarez:28
9184011-1:Laura Andrade:47
12028339-1:Jorge Argandoña:29
10523653-0:Camila Avaria:40
12187197-1:Felipe Ávila:36
5935556-2:Aquiles Barriga:80
14350739-4:Eduardo Bello:29
6951420-0:Cora Benítez:68
11370775-5:Hugo Berger:31
11111756-k:Cristóbal Bórquez:34
```

Además, cada vez que alguien se atiende en la consulta, la visita es registrada en el archivo `atenciones.txt`, agregando una línea que tiene el rut del paciente, la fecha de la visita (en formato `dia-mes-año`) y el precio de la atención, también separados por `:`. El archivo se ve así:

```
8015253-1:4-5-2010:69580
12393241-2:6-5-2010:57274
10985778-5:8-5-2010:73206
8015253-1:10-5-2010:30796
8015253-1:12-5-2010:47048
12028339-1:12-5-2010:47927
11426761-9:13-5-2010:39117
10985778-5:15-5-2010:86209
7752666-8:18-5-2010:41916
8015253-1:18-5-2010:74101
12187197-1:20-5-2010:38909
8654231-5:20-5-2010:75018
8654231-5:22-5-2010:64944
```

```
5106360-0:24-5-2010:53341
8015253-1:27-5-2010:76047
9217890-0:30-5-2010:57726
7752666-8:1-6-2010:54987
8509454-8:2-6-2010:76483
6092377-9:2-6-2010:62106
11370775-5:3-6-2010:67035
11370775-5:7-6-2010:47299
8509454-8:7-6-2010:73254
8509454-8:10-6-2010:82955
11111756-k:10-6-2010:56520
7752666-8:10-6-2010:40820
12028339-1:12-6-2010:79237
11111756-k:13-6-2010:69094
5935556-2:14-6-2010:73174
11111756-k:21-6-2010:70417
11426761-9:22-6-2010:80217
12067539-7:25-6-2010:31555
11370775-5:26-6-2010:75796
10523653-0:26-6-2010:34585
6951420-0:28-6-2010:45433
5106360-0:1-7-2010:48445
8654231-5:4-7-2010:76458
```

Note que las fechas están ordenadas de menos a más reciente, ya que las nuevas líneas siempre se van agregando al final.

1. Escriba la función `costo_total_paciente(rut)` que entregue el costo total de las atenciones del paciente con el rut dado:

```
>>> costo_total_paciente('8015253-1')
297572
>>> costo_total_paciente('14350739-4')
0
```

2. Escriba la función `pacientes_dia(dia, mes, ano)` que entregue una lista con los nombres de los pacientes que se atendieron el día señalado:

```
>>> pacientes_dia(2, 6, 2010)
['Pablo Muñoz', 'Alfonso San Martín']
>>> pacientes_dia(23, 6, 2010)
[]
```

3. Escriba la función `separar_pacientes()` que construya dos nuevos archivos:
 - `jovenes.txt`, con los datos de los pacientes menores de 30 años;
 - `mayores.txt`, con los datos de todos los pacientes mayores de 60 años.

Por ejemplo, el archivo `jovenes.txt` debe verse así:

```
15007265-4:Andrés Morales:26
15690109-1:Francisco Ruiz:26
13087677-3:Jorge Álvarez:28
12028339-1:Jorge Argandoña:29
14350739-4:Eduardo Bello:29
```

4. Escribir una función `ganancias_por_mes()` que construya un nuevo archivo llamado `ganancias.txt` que tenga el total de ganancias por cada mes en el siguiente formato:

```
5-2010:933159
6-2010:1120967
7-2010:124903
```

Inventario

Una tienda tiene la información de sus productos en un archivo llamado `productos.txt`. Cada línea del archivo tiene tres datos:

- el código del producto (un número entero),
- el nombre del producto, y
- la cantidad de unidades del producto que quedan en bodega.

Los datos están separados por un símbolo `/`. Por ejemplo, el siguiente puede ser el contenido del archivo:

```
1265/Reloj/26
613/Cuaderno/87
9801/Vuvuzela/3
321/Lápiz/12
5413/Tomate/5
```

1. Escriba la función `existe_producto(codigo)` que indique si existe el producto con el código dado:

```
>>> existe_producto(1784)
False
>>> existe_producto(321)
True
>>> existe_producto(613)
True
>>> existe_producto(0)
False
```

2. Escriba la función `por_reabastecer()` que cree un nuevo archivo llamado `por_reabastecer.txt` que contenga los datos de todos los productos de los que queden menos de 10 unidades.

En este caso, el archivo `por_reabastecer.txt` debe quedar así:

```
9801/Vuvuzela/3
5413/Tomate/5
```

Donantes

Una institución de beneficencia tiene un registro de las personas que han hecho donaciones en un archivo de registros llamado `donantes.txt`

El archivo está ordenado por rut de menor a mayor. Para simplificar, vamos a suponer que los ruts tienen cinco dígitos, y no incluyen el dígito después de la raya.

Por ejemplo, el contenido del archivo puede ser el siguiente:

Rut	Nombre	Monto
15274	Fulana de Tal	200
15891	Jean Dupont	150
16443	Erika Mustermann	400
16504	Perico Los Palotes	80
17004	Jan Kowalski	200

Los problemas son los siguientes:

1. Escribir una función que cree el archivo con los datos de la tabla.
2. Escribir una función que muestre los datos del archivo.
3. Escribir una función que pida al usuario ingresar un rut, y muestre como salida el monto donado por esa persona.
4. Escribir una función que pida al usuario ingresar un rut, y elimine del archivo al donante con ese rut.
5. Escribir un programa que pida al usuario ingresar los datos de un donante, y los agregue al archivo.

Mezcla de números

Los archivos `a.txt` y `b.txt` tienen muchos números ordenados de menor a mayor.

Escriba un programa que cree un archivo `c.txt` que contenga todos los números presentes en `a.txt` y `b.txt` y que también esté ordenado.

No guarde los números en una estructura de datos. Vaya leyéndolos y escribiéndolos uno por uno.