

A los empleados de una empresa multinacional que realizan viajes de negocios les pagan los gastos de dichos viajes. Después de cada viaje, deben cargar los tickets a través de una aplicación. Dicha aplicación genera un archivo `rendicion_gatos.csv`, que contiene los siguientes datos:

`Legajo_empleado, fecha_ticket(AAAAMMDD), id_ticket, cod_reintegro, monto`

El archivo se encuentra ordenado por `legajo_empleado`, `cod_reintegro` y no puede ser cargado en memoria. Es necesario escribir un programa modular en Python que procese dicho archivo para:

1. Informar mediante un listado cuanto se debe devolver a cada empleado por tipo de gasto (`cod_reintegro`), considerando que existen topes para cada código de reintegro. Dicho tope, se encuentra en un archivo que puede ser cargado en su totalidad en memoria, con la función `cargar_topes()` que se encuentra en el módulo `herramientas.py`, no debe desarrollarlo, simplemente utilizarlo. La función devuelve un objeto diccionario, del tipo `{cod_reintegro: [descripcion, tope],,,,,,}`
En caso, que el empleado haya excedido el tope, debe enviarse a un archivo `excedidos.txt`, el mensaje correspondiente, indicando el legajo del empleado, el código de reintegro, la descripción del código, y el importe por el cual se excedió.
2. Generar un archivo `ranking_devoluciones.csv`, ordenado por monto total reintegrado, de mayor a menor; que contenga, el número de legajo, y el monto correspondiente. Evitar recorrer el archivo `rendicion_gastos.csv` nuevamente.

Notas:

- Desarrollar el código en Python, en forma estructurada, aplicando funciones
- Todos los archivos existirán.
- La fecha y el `id_ticket` se pueden trabajar como strings
- Escribir con letra clara
- No usar archivos auxiliares
- Minimizar los accesos a las estructuras

A los empleados de una empresa multinacional que realizan viajes de negocios les pagan los gastos de dichos viajes. Después de cada viaje, deben cargar los tickets a través de una aplicación. Dicha aplicación genera un archivo `rendicion_gatos.csv`, que contiene los siguientes datos:

`Legajo_empleado, fecha_ticket(AAAAMMDD), id_ticket, cod_reintegro, monto`

El archivo se encuentra ordenado por `legajo_empleado`, `cod_reintegro` y no puede ser cargado en memoria. Es necesario escribir un programa modular en Python que procese dicho archivo para:

1. Informar mediante un listado cuanto se debe devolver a cada empleado por tipo de gasto (`cod_reintegro`), considerando que existen topes para cada código de reintegro. Dicho tope, se encuentra en un archivo que puede ser cargado en su totalidad en memoria, con la función `cargar_topes()` que se encuentra en el módulo `herramientas.py`, no debe desarrollarlo, simplemente utilizarlo. La función devuelve un objeto diccionario, del tipo `{cod_reintegro: [descripcion, tope],,,,,,}`
En caso, que el empleado haya excedido el tope, debe enviarse a un archivo `excedidos.txt`, el mensaje correspondiente, indicando el legajo del empleado, el código de reintegro, la descripción del código, y el importe por el cual se excedió.
2. Generar un archivo `ranking_devoluciones.csv`, ordenado por monto total reintegrado, de mayor a menor; que contenga, el número de legajo, y el monto correspondiente. Evitar recorrer el archivo `rendicion_gastos.csv` nuevamente.

Notas:

- Desarrollar el código en Python, en forma estructurada, aplicando funciones
- Todos los archivos existirán.
- La fecha y el `id_ticket` se pueden trabajar como strings
- Escribir con letra clara
- No usar archivos auxiliares
- Minimizar los accesos a las estructuras