

Universidad Nacional de Asunción - Facultad Politécnica

Primer Examen Final de Algoritmo

Fecha: 30 de noviembre de 2024

Directivas y Recomendaciones

- Los comentarios aclaratorios pueden ayudar en la corrección del examen, influyendo favorablemente en su calificación.
- La duración del examen es de 120 minutos.
- El código fuente para cada tema debe subirse al aula virtual de la materia (en EDUCA), en la actividad correspondiente en el VPL.

Ejercicio 1: Procesar planilla de salarios (100%)

Como parte del procesamiento de los salarios de los empleados de una empresa, resulta necesario implementar en Python una función `obtenerDatosSalario()`, que reciba una cadena (que contiene los datos del salario de un empleado) con el siguiente formato:

CantidadMoneda

Donde **Cantidad** representa un número entero (empleándose el carácter punto '.' como separador de miles); mientras que **Moneda** representa la moneda de pago (por ejemplo: **Gs** se refiere a guaraníes, **\$** a dólares, **R\$** a reales, **PArg** a peso argentino, etc.).

Algunos ejemplos de las cadenas con este formato son:

- 1.234.567Gs
- 5.670\$
- 20.000R\$
- 158Programacion
- 1.000.000.000PArg

La función `obtenerDatosSalario()` debe retornar la siguiente información:

- **Cantidad** (número entero): donde se almacenará el monto percibido por el empleado.
- **Moneda** (cadena): donde se almacenará el nombre de la moneda. Contendrá:
 - "Guaranies" si **Moneda** es **Gs** (sin diferenciar letras mayúsculas/minúsculas)
 - "Dolares" si **Moneda** es **\$**
 - "Reales" si **Moneda** es **R\$** (sin diferenciar letras mayúsculas/minúsculas)
 - "Otro" para cualquier otro caso en **Moneda**

Con la ayuda de la función `obtenerDatosSalario()`, se debe procesar una planilla de empleados que se encuentra en el archivo de entrada `empleados.txt`. Este archivo tiene los siguientes campos (separados por tabuladores):

Nombre Apellido Salario Dependencia

Por ejemplo:

| | Nombre | Apellido | Salario | Dependencia |
|---|--------|----------|-------------|-------------|
| 1 | Jose | Colbes | 6.300.000gS | FPUNA |
| 2 | Andrea | Insfran | 3.500\$ | FIUNA |
| 3 | Juan | Perez | 5.000r\$ | CEMIT |
| 4 | Maria | Cuevas | 100.000PArg | FADA |

A partir del archivo de entrada, se debe generar un archivo de salida llamado **empleados_Mod.txt**. La primera línea del archivo debe contener la descripción de sus campos, que son los siguientes:

Nombre Apellido Cantidad Moneda Dependencia

Y en las siguientes líneas se deberán tener los datos correspondientes a cada empleado (donde los campos también están separados por tabuladores). Nótese que los campos **Cantidad** y **Dependencia** corresponden a lo retornado por la función **obtenerDatosSalario()**, a partir del campo **Salario** del archivo de entrada.

Para el ejemplo de entrada, se tendrá el siguiente archivo de salida:

| 1 | Nombre | Apellido | Cantidad | Moneda | Dependencia |
|---|--------|----------|----------|-----------|-------------|
| 2 | Jose | Colbes | 6300000 | Guaranies | FPUNA |
| 3 | Andrea | Insfran | 3500 | Dolares | FIUNA |
| 4 | Juan | Perez | 5000 | Reales | CEMIT |
| 5 | Maria | Cuevas | 100000 | Otro | FADA |
| 6 | | | | | |

Observaciones importantes

- Se puede suponer que, en el campo de texto que representa al salario (dentro del archivo) siempre se tendrá una cadena en el formato especificado.
- Se puede suponer que **Cantidad** siempre representará a un número entero con el formato de separador de miles especificado (es decir, no hace falta verificar esto).
- Tener en cuenta que **Moneda** puede contener cualquier información, pero se puede suponer que no contendrá caracteres que representen a números.

Criterios de corrección (Total: 100%):

- Lectura de datos de entrada: 15%
- Sintaxis y comentarios: 10%
- Implementación y uso de la función **obtenerDatosSalario()**: 35%
- Procesamiento de la planilla de empleados: 10%
- Generación del archivo de salida con el formato especificado: 15%
- Uso de funciones para dividir el problema en partes (al menos dos funciones además de **obtenerDatosSalario()**): 15%