Universidad de Santiago de Chile
Facultad de Administración y Economía
Contador Público y Auditor

FUNDAMENTOS DE PROGRAMACIÓN

Enunciado Trabajo

Santiago, junio de 2017

I. Objetivos

Aplicar los conocimientos de Programación vistos durante el curso para generar un programa en Python que aplique los contenidos vistos hasta el momento.

II. Problema

Una voz misteriosa le dijo a Noé que construyera un bote gigante y pusiera una pareja de cada especie animal existente en dicho bote, pues el calentamiento global hará que las mareas suban y la tierra será cubierta de agua.

Con el arca ya construida, Noé y sus asociados, han comenzado la búsqueda de un macho y una hembra de cada especie para completar la tarea que la voz misteriosa les asignó. Para evitar problemas de entendimiento la voz, les entregó un archivo "*listado.csv*" con la lista de los animales que necesitan conseguir, indicando:

- ID_ANIMAL: Número que se le da a una especie de animal para diferenciarla de otras especies.
- NOMBRE_ANIMAL: Nombre con el que se conoce comúnmente al animal en español
- NOMBRE_CIENTÏFICO: Nombre con el que se denomina al animal en la comunidad científica.

Un ejemplo del archivo "*listado.csv*" se puede ver a continuación:

```
ID_ANIMAL,NOMBRE_ANIMAL,NOMBRE_CIENTIFICO
1,Chincilla de cola larga, Chinchilla lanígera
2,Perro, Canis lupus familiaris
3,Tigre Siberiano, Panthera tigris virgata
…
1002,Jirafa,Giraffa camelopardalis
```

Por otro lado, Noé quién es una persona muy ordenada, obliga a sus asociados a llenar un registro con los animales que ya han conseguido, por lo que estos datos quedan almacenados en un único archivo llamado "*registro.csv*", indicando:

- ID_ANIMAL: Identificador de la especie de animal capturada, para evitar escribir los nombres.
- NOMBRE: Nombre propio del animal, dos animales de una misma especie no pueden tener el mismo nombre, pero podrían haber animales de especies distintas que lo compartan.
- SEXO: Sexo del animal.

Un ejemplo del archivo "registro.csv" puede verse a continuación:

```
ID_ANIMAL,NOMBRE,SEXO
1, Pepita la Chinchilla, Hembra
2, Blacky, Macho
2, Osa, Hembra
2, DaCorgi, Macho
…
1024, Dumbo, Macho
```

Sin embargo, con estos dos registros que Noé ha revisado meticulosamente, se ha dado cuenta de que aún persisten tres problemas:

- Él y sus asociados no saben que parejas de animales ya están completas, por lo que a menudo terminan con más de un macho o una hembra de una especie y por ello gastan tiempo capturando y liberando animales que no necesitan. Por lo que necesitan una forma de saber que animales liberar.
- Al no tener claras que parejas de animales ya están completas, siguen buscando animales que ya tienen. Por lo que necesitan saber que animales dejar de buscar y cuáles están pendientes.

- Al no tener claro que animal de la pareja ya tienen (macho o hembra), Noé no puede realizar su búsqueda de animales de la forma más efectiva, pues constantemente debe revisar si al capturar un macho, ya hay uno, y lo mismo para las hembras.

Por ello, Noé necesita que usted genere un programa en Python que resuelva sus problemas de datos de las siguientes formas:

Requiere actualizar el "*listado.csv*" con una columna extra llamada "ESTADO" que indique el estado de la especie, es decir, si ya está la pareja, si falta un macho o una hembra, o ambos, por ejemplo, un "*nuevoListado.csv*" debería verse así:

```
ID_ANIMAL,NOMBRE_ANIMAL,NOMBRE_CIENTIFICO;ESTADO
1,Chincilla de cola larga, Chinchilla lanígera, FALTA UNA HEMBRA
2,Perro, Canis lupus familiaris, OK
3,Tigre Siberiano, Panthera tigris virgata, FALTA UN MACHO
…
1002, Jirafa, Giraffa camelopardalis, FALTAN DOS
```

En segundo lugar, como el espacio en su barca, es limitado, necesita saber que animales liberar, para ello, se requiere un nuevo archivo "*condenados.csv*" que indica que animales no deben llevar en el bote y que deben ser puestos en libertad inmediatamente:

```
ID_ANIMAL;NOMBRE;SEXO;NOMBRE_CIENTIFICO
2, Blacky, Macho, Canis lupus familiaris
5, Ramona, Hembra, Felis silvestris catus
16, Pumba, Macho, Sus scrofa
```

Considere que ambos archivos deben ser salidas nuevas y deben generarse por un solo programa en Python.

II.1. Requisitos previos

El trabajo solicitado debe desarrollarse en Python, en la versión utilizada en el curso, es decir Python 2.7.

II.2. Respecto a la representación en Python

Considere que este trabajo fue conscientemente diseñado para ser resuelto utilizando únicamente los contenidos y bibliotecas (librerías) vistos hasta el momento, por lo que no debe ser necesario investigar o añadir otros complementos adicionales para ejecutar el código.

Considere que se evaluará la estructura de programas en Python por lo que, a pesar de la envergadura del programa, éste **DEBE** estar subdividido en funciones claramente definidas y comentadas, indicando la tarea realizada, sus entradas y salidas. De ser necesario, se solicita incorporar comentarios paso a paso para las instrucciones.

Los bloques de programa deben ir claramente definidos, y como mínimo debiera diferenciar:

- Encabezado: Indicado la información relevante del programa.
- Definición de constantes: Indicando la importación y definición de valores constantes en el programa.
- Definición de funciones: Indicando las funciones necesarias para la ejecución del programa, con sus respectivas entradas, salidas y la labor que realizan.
- Bloque Principal:
    - Entrada: Solicitando las entradas necesarias para el funcionamiento del programa.
    - Procesamiento: Realizando las funciones necesarias para la generación de la salida
    - Salida: Notificando al usuario de las acciones que se han realizado y la respuesta generada.

Considere que identificando y dividiendo el programa en sub-procesos, el problema resultará más sencillo de abordar para el equipo de trabajo, en caso de alcanzar sólo parcialmente los objetivos

del control, se sugiere colocar en los comentarios las abstracciones de procesos tanto para aquellos implementados como no implementados.

III. Entrega

El plazo máximo para entregar el programa será el día viernes 1 de julio a más tardar a las 18:00 horas.

Se requiere entregar un archivo .py con el programa desarrollado en Python, con los RUT de sus integrantes (sin puntos, ni guión) por nombre de archivo siguiendo el formato <RUT N°1>-<RUT N°2>.py, por ejemplo:

173454322-18984321K.py

Además, en el código se debe añadir el siguiente encabezado del programa, con los siguientes datos para identificar su trabajo (rellene el encabezado del programa con los datos en formato IDÉNTICO al indicado en el ejemplo a continuación):

```
# -*- coding: cp1252 -*-

# SECCIÓN: B-1
# PROFESOR: JUAN GONZÁLEZ

# INTEGRANTES
      # 1.
      # NOMBRE: Juan Carlos Perez Gonzalez
      # RUT: 17.345.432-2

      # 2.
      # NOMBRE: Javier Rojas Madariaga
      # RUT: 18.984.321-K

      # 3.
      # NOMBRE: Macarena Javiera Cavieres Maldonado
      # RUT: 18.385.223-7


# DESCRIPCIÓN DEL PROGRAMA ...<CONTINÚE CON EL PROGRAMA A PARTIR DE AQUÍ>
```

Se requiere que envíe este archivo al profesor del curso a través de la intranet institucional y a los medios que su profesor disponga como respaldo **ANTES** del plazo establecido. En caso de que existan errores en la plataforma, enviarlo al correo electrónico del profesor.

En caso de tener más archivos en su implementación, considere que el encabezado debe ser parte de cada uno de los archivos y se debe entregar el trabajo completo en un archivo comprimido de extensión ".zip" o ".rar" con el mismo nombre que se indicó anteriormente.
Considere que archivos sin identificador de autores tanto en el nombre como dentro del encabezado del código, **no serán revisados, ni evaluados**.

Del mismo modo considere que archivos enviados únicamente al correo del profesor, sin copia en intranet, no serán considerados en la revisión.

IV. Evaluación

Respecto a la evaluación tenga en consideración las siguientes instrucciones (El no seguimiento de una o más de estas normas puede significarle ser evaluado con la nota mínima):

- Los grupos estarán compuestos por un máximo de **tres personas**, ambas pertenecientes a la **misma sección del curso**.
- Considere que los archivos ilustrados en el enunciado, o entregados por los profesores corresponden a EJEMPLOS y que el programa debe funcionar para archivos de cualquier tamaño, siempre y cuando cumplan con el formato establecido.
- Se permite el trabajo individual, sin embargo, esto no significa una disminución en los criterios de evaluación.
- En caso de detectar **programas iguales, o con suficientes similitudes entre sí**, tanto en una misma sección como en secciones distintas, será considerado un acto de copia y calificado con nota mínima. Se compararán programas con un algoritmo de reconocimiento de texto y además se pasará por un proceso de validación visual por los profesores del curso para realizar dicha detección.
- Grupos que entreguen fuera del plazo establecido serán calificados con la nota mínima.

- Grupos que sean descubiertos en actos deshonestos serán sancionados con la nota mínima.
- Considere que la implementación de su programa es de responsabilidad **ÚNICA** de los integrantes del grupo, y que se han entregado todas las herramientas y los contenidos para el desarrollo de esta, por lo que **cualquier indicio de intervención de un tercero**, será sancionado con la nota mínima.
- La difusión de este enunciado, solicitando ayuda, ofreciendo dinero a cambio de la resolución e incluso la publicación de éste dentro del plazo de desarrollo de la tarea, se considera un acto deshonesto, que daña la imagen de la Facultad de Administración y Economía y de la Universidad de Santiago por lo que en caso de detectarse este caso se entregarán los antecedentes a la Universidad para que tome las acciones pertinentes al caso.
- Se sancionará con descuentos sobre la calificación final a los grupos que no cumplan con las instrucciones del programa y de entrega publicados en el enunciado.
- Se bonificará a los grupos que añadan funcionalidades adicionales, o aspectos de presentación más elaborados y que en general no se limiten simplemente a entregar el mínimo para la calificación (en estos casos indicar en los comentarios o en un archivo .txt, requisitos, modo de uso, funcionalidades extra, etc).
- Considere que se evaluará el código mismo, las funcionalidades conseguidas, la estructura del programa, la ejecución del programa, **la respuesta a eventos inesperados** (por ejemplo al ingresar valores inválidos) y los comentarios en el código fuente.
- Cualquier situación no contemplada en este documento será dirimida por la Facultad de Economía.

La nota final de esta evaluación se divide en dos partes, la implementación y el test de salida con las siguientes características:

- El test de salida corresponde a un control que será tomado la clase siguiente a la entrega del enunciado, que corresponde a una interrogación respecto al funcionamiento del código, a fin de verificar que el grupo que presenta un trabajo que es de su autoría, en el cuál se le preguntará respecto a aspectos funcionales de **SU** implementación o la resolución sin el intérprete de Python de alguno de los sub-problemas del enunciado.

- Si su nota del test es superior a 4,0, la ponderación es 10% test de salida y 90% programa.
- Si su nota en el test de salida es inferior a 4,0, la ponderación del trabajo es 100% la nota más baja entre el control de salida y la nota del programa.

- En caso de no asistir al test de salida, si el estudiante no está justificado, este será calificado con la nota mínima.