

Laboratorio 3 - Funcional

INS125 - Leguajes de programación
Universidad Andrés Bello

23 de mayo de 2020

1. Situación

Debido a la actual crisis sanitaria, se requiere tener un mayor control en el acceso a las muestras extraídas a diversos pacientes en el prestigioso laboratorio de la Universidad Andrés Bello. Por esta razón, el director del laboratorio ha exigido que quienes deban ingresar porten un código especial de acceso.

Se le ha solicitado a usted desarrollar un programa que permita generar el código de acceso para cada uno de los laboratoristas y tecnólogos médicos encargados de procesar las muestras.

2. Cifrado

El código de acceso corresponde a un mensaje cifrado que se genera con la información de cada usuario (nombre, apellido, rut). Usted deberá leer la información y manipularla para obtener el código. Utilice el nombre y apellido del usuario para generar una cadena. Deberá modificar los elementos de la cadena desplazándolos una determinada cantidad de posiciones dentro del alfabeto, dicha cantidad corresponde al dígito verificador del rut.

Ejemplo: Juan Ramos 17.232.415-2

En este ejemplo la cadena sería: 'JuanRamos' y al aplicar la modificación, la cadena resultante sería "LwcpTcoqu", debido a que sus caracteres se desplazaron 2 posiciones, ya que el dígito verificador de su rut es 2.

Una vez haya obtenido su nueva cadena, deberá procesarla para obtener los valores *ascii* de cada carácter, aplicar las restricciones que se indiquen, volver a convertirla en caracteres y agregarle un código validador.

3. Tarea

Comprendiendo lo anterior se le solicita realizar lo indicado siguiendo las siguientes indicaciones:

1. Lea la lista de usuarios del archivo `usuarios.csv`, la cual contiene el nombre, apellido, rut y fecha de nacimiento, separados por un caracter de punto y coma `;`. Para eso deberá utilizar la función `leerArchivo()` que ya está implementada en el código base.
2. Filtre los usuarios que posean una edad inferior a 60 años (nacidos antes del 01-01-1960). Los usuarios con edad mayor o igual a 60 años, es decir, aquellos nacidos el año 1 de enero del año 1960 o antes, no podrán ingresar al recinto por pertenecer al grupo de riesgo. Guarde un archivo con registros separados por `;` llamado `riesgo.csv` donde se encuentren los usuarios que no tienen permitido el ingreso y otro de las mismas características llamado `permitidos.csv` el cual contendrá la lista con los usuarios que no pertenecen al grupo de riesgo. Para esto debe utilizar la función `filter` de la biblioteca `functools`.
3. Con los usuarios que no pertenecen al grupo de riesgo (es decir, ningún usuario presente en el archivo `riesgo.csv`), agrupe el nombre y apellido del usuario en una sola cadena y continúe con los pasos siguientes.
4. Implemente el cifrado inicial con la cadena obtenida anteriormente y con un desplazamiento equivalente al dígito verificador del rut del usuario. Considere que el dígito verificador `k` corresponde a una rotación de 1. También tenga en cuenta que los caracteres que tengan tildes deben ser reemplazados por su equivalente sin tilde (es decir, el carácter `á` debe ser reemplazado por `a`).
5. Convierta la nueva cadena a valores numéricos.
6. Calcule la suma de dichos valores (considere cada elemento de la cadena como UN número).
7. Cree una lista basándose en la cadena de valores numéricos, donde la lista contendrá el valor de la cadena separada en elementos de dos números.
8. Su lista deberá ser traducida al carácter que corresponda para cada par de números de la lista generada anteriormente.
9. El código de acceso del usuario será el resultado de la lista anterior concatenada con un código validador, dicho código se obtiene con la función `validador()`.
10. Finalmente, usted deberá generar un archivo llamado: `'info.csv'`, donde deberá escribir el nombre, apellido, rut y código de acceso para cada usuario separados por un `;`.

La función `validador()` entrega el resultado de la suma obtenida en el punto 6 multiplicada por el dígito verificador del rut del usuario.

4. Restricciones

- Debe implementar las funciones (como funciones con identificador):

- leerArchivo(nombrearchivo)
- cifradoinicial(string,rot)
- sumaCod(string)
- separadorCifrado(string,rut)
- validarMenores(lista)
- validarEspeciales(lista,especiales)
- validador(suma,rut)

Todas las demás funciones que utilice su programa deben ser funciones anónimas (*Lambda*) utilizando el paradigma funcional.

- Para desarrollar su solución deberá utilizar las funciones map, reduce y filter.
- En el punto 7, cuando obtenga la lista de pares de números, debe considerar que dicho valor (del par) no puede ser inferior a 35, en caso de serlo, dicho valor será modificado por su valor más 35.
- Además de lo anterior hay una lista de caracteres que al director del laboratorio le desagradan, por lo cual usted deberá reemplazarlos en caso de encontrarlos. Para eso se le entregará una lista que contiene los caracteres que usted debe reemplazar en caso de que aparezcan. Para reemplazarlos sólo deberá sumar 10 al valor encontrado.
- Si en el punto 7 su cadena de valores numéricos tiene un largo impar, agregue el dígito verificador al final de ella.
- Debe considerar el alfabeto inglés para el cifrado inicial. No considere tildes.
- En caso de encontrar el caracter 'ñ' reemplácelo por una 'n'.
- Si el rut verificador terminase en 'k', reemplácelo por un 1.
- No está permitido hacer más imports de los ya existentes. Es decir, sólo puede utilizar las bibliotecas `string`, `functools`

5. Instrucciones

- Fecha de entrega: Sábado 30 de mayo, 2020 a las 23:59.
- Trabajo personal hecho en lenguaje Python.
- Método de entrega: Su repositorio privado creado a través del link de la tarea. **No se aceptaran entregas en repositorios no creados a través del link de la tarea.**
- Para comenzar su tarea, clone su repositorio y utilice el archivo `main.py`, con código pre hecho. Ese código le servirá para poder leer el archivo de entrada y utilizar los módulos `ascii_lowercase`

6. Recomendaciones

- Se recomienda hacer commits parciales. Si sus archivo de salida no contiene toda la matriz o el resultado completo de la Convolución pero si contiene parcialmente la matriz, puede obtener puntaje parcial.
- Si su programa no crea el o los archivos de salida, o no se ejecuta, será evaluado con la nota mínima.
- En el repositorio oficial del laboratorio puede encontrar dos ejemplos de archivos de entrada y su correspondiente salida. <https://github.com/INS125/Laboratorio/>
- Recuerde solicitar unirse a github student. Si no lo hace, no podrá hacer su código privado y cualquiera podría visualizar su tarea.
- Es un trabajo personal. Es su responsabilidad cuidar su tarea.

7. Código de honor

Toda persona inscrita en este curso se compromete a:

- Actuar con honestidad, rectitud y buena fe frente a sus profesores y compañeros.
- No presentar trabajos o citas de otras personas como propias o sin su correspondiente citación, ya sea de algún compañero, libro o extraídos de internet como también a no reutilizar trabajos presentados en semestres anteriores como trabajos originales.
- No copiar a compañeros ni hacer uso de ayudas o comunicaciones fuera de lo permitido durante las evaluaciones.

Cualquier alumno o alumna que no respete el código de honor durante una evaluación (sea este la entrega de una tarea o el desarrollo de una prueba o control tanto durante la cátedra como el laboratorio) será evaluado con la nota mínima y será virtud de profesor, de acuerdo con la gravedad de la falta, las acciones siguientes a tomar.

8. Anexo

8.1. 'Ejemplo: Código de acceso'

Si consideramos la siguiente entrada:

['Jonathan','Kreisberg','16111111-2','13-10-1980']

Usted deberá validar si el usuario es apto para ingresar al laboratorio. Como en este caso su edad es menor a 60 años (es decir, nació después del 01-01-1960), se continuará el procedimiento.

1. **Concatenar el texto**

JonathanKreisberg

2. **Aplicar cifrado inicial**

LqpcvjcpMtgkudgti

3. **Convertir Cifrado a Entero**

76113112991181069911277116103107117100103116105

4. **Separar entero en lista de pares**

['76','11','31','12','99','11','81','06','99','11','27','71','16','10','31','07','11','71','00','10','31','16','10','52']

5. **Modificar elementos de lista (menores a 35)**

['76','46','66','47','99','46','81','41','99','46','62','71','51','45','66','42','46','71','35','45','66','51','45','52']

6. **Modificar caracteres especiales** *['76','56','66','57','99','56','81','51','99','56','72','71','51','55','66','52','56','71','35','55','66','51','55','52']*

7. **Transformar a caracteres**

L8B9c8Q3c8HG37B48G7B374

8. **Antes de generar el código final recuerde calcular la operación señalada en el enunciado (*operación validador()*)**

Cifrado: (76113112991181069911277116103107117100103116105)=[141]

Dígito verificador: (16.111.111-2)=[2]

Resultado (141 * 2): 282

9. **Código de acceso**

L8B9c8Q3c8HG37B48G7B374282

10. **Salida de su programa**

Jonathan;Kreisberg;16.111.111-2;**L8B9c8Q3c8HG37B48G7B374282**