

# Programación – Certamen 1 - Miércoles 28 de Octubre de 2020

## Instrucciones

Este certamen está compuesto por 3 preguntas de desarrollo y 1 video.

Usted dispondrá de 150 minutos para completar las preguntas de desarrollo y enviarlas a través de los buzones habilitados en Aula, esto es, hasta las 19:50. El video podrá ser entregado hasta las 21:00 horas, subiendo el enlace respectivo en un buzón de Aula habilitado para ese efecto.

Suba su video a *YouTube* o a *Google Drive*, según las instrucciones enviadas previamente. Si utiliza *YouTube*, asegúrese de que quede **Oculto**, pero por ningún motivo público o privado. Si utiliza *Google Drive*, asegúrese de dejarlo visible para quien tenga el *link*. No publique su video por ningún otro medio o red social.

## Entrega

Estarán disponibles 4 buzones de entrega: uno para cada una de las preguntas y uno para que entregue ahí el enlace al video.

En el caso de las preguntas 1 y 3, debe subir el archivo con el código de su solución, es decir, con **extensión .py**. No suba archivos comprimidos (.zip). En caso de utilizar *repl.it*, que descarga un archivo .zip, asegúrese de descomprimirlo y enviar únicamente el archivo .py.

Para el caso del ruteo de la Pregunta 2, suba el archivo Excel, pdf o la imagen con su solución.

## Instrucciones relacionadas al video

Grabe **un único video**, cuya duración debe ser de entre 3 y 5 minutos, explicando la forma en que resolvió únicamente la **pregunta 3**.

**NO** debe grabarse escribiendo el código, sino que una vez que lo finalice, comente las estrategias que utilizó para resolver el problema.

Para orientarle en la grabación del video, le recomendamos guiarse con las siguientes preguntas:

- ¿Cómo realizó la separación del RUT y el dígito verificador, para hacer la validación requerida?
- ¿Cuál es el algoritmo que implementó para determinar el local de votación, según el apellido?
- ¿Cómo determina su programa el número de mesa que corresponde a una persona?

## Ponderación del video

Es importante que grabe y envíe el video incluso si la pregunta 3 está parcialmente completa. Esto, porque el video es un ponderador sobre la nota de esta pregunta:

Sea  $nd$  la nota obtenida en el desarrollo de la pregunta 3, antes de la revisión del video:

- Si no adjunta video,  $nota\_pregunta = nd \times 0 = 0$
- Si adjunta el video, y
  - \* Explica correctamente su código en la grabación y demuestra comprensión de ciclos, condicionales y strings,  $nota\_pregunta = nd \times 1 = nd$
  - \* Hay un esfuerzo por darse a entender en la grabación, y evidencia comprensión de ciclos, condicionales y strings,  $nota\_pregunta = nd \times 0,7$
  - \* La explicación grabada consiste en una lectura del código, sin profundizar en su razonamiento,  $nota\_pregunta = nd \times 0,3$

# Programación – Certamen 1 - Miércoles 28 de Octubre de 2020

## Contexto

El dígito verificador, que aparece después del guión en todo RUT, permite detectar errores de digitación o transcripción. Se obtiene a partir de los dígitos que aparecen antes del guión, multiplicando cada uno de ellos por una constante asociada con la posición en que se encuentra, para luego obtener la suma  $p$  de estos productos. Comenzando desde la derecha, las constantes que se debe utilizar son 2, 3, 4, 5, 6, 7, 2, 3. Luego, se toma el resto de dividir  $p$  entre 11, que llamaremos  $r$ . Para obtener el dígito verificador se calcula  $11 - r$ . Si el resultado es 11, entonces se convierte en 0. Si resulta ser 10, entonces se convierte en la letra  $k$ .

Ejemplos:

Si el RUT, antes del guión, es 13252311, se calcula  $p$ , aplicado las constantes, de la siguiente manera:

$$p = (1 \times 3) + (3 \times 2) + (2 \times 7) + (5 \times 6) + (2 \times 5) + (3 \times 4) + (1 \times 3) + (1 \times 2) = 80$$

$$\text{verificador} = 11 - (80 \% 11) = 11 - 3 = 8$$

Entonces, el RUT completo es: 13252311-8.

Si el RUT, antes del guión, es 15136120, se calcula  $p$ , aplicado las constantes, de la siguiente manera:

$$p = (1 \times 3) + (5 \times 2) + (1 \times 7) + (3 \times 6) + (6 \times 5) + (1 \times 4) + (2 \times 3) + (0 \times 2) = 78$$

$$\text{verificador} = 11 - (78 \% 11) = 11 - 1 = 10 \rightarrow k$$

Entonces, el RUT completo es: 15136120-k.

Usted cuenta con la función **suma\_productos(rut)** que calcula la suma de productos  $p$ . El parámetro de la función es un número entero que corresponde a un RUT.

```
def suma_productos(rut):
    suma_n = 0
    i = 2 # constante a multiplicar, comenzando por la derecha: 2, 3, 4, 5, 6, 7, 2, 3
    while rut != 0:
        n = rut % 10 # obtiene el dígito de mas a la derecha
        rut = rut // 10 # elimina el dígito de mas a la derecha
        suma_n += n*i
        # A continuación, se actualiza la constante para la siguiente iteración
        i += 1
        if i == 8: # después del 7, vuelve a 2
            i = 2
    return suma_n
```

1. [20 %] Escriba la función **digito\_verificador(rut)**, que recibe como parámetro un número entero que representa un RUT. La función debe calcular y retornar el dígito verificador correspondiente. Para ello, es obligatorio utilizar correctamente la función **suma\_productos** suministrada.

Ejemplo:

```
>>> print(digito_verificador(13252311))
8
>>> print(digito_verificador(15136120))
k
```

## Programación – Certamen 1 - Miércoles 28 de Octubre de 2020

2. [30 %] Haga un ruteo para el siguiente programa, ingresando como entrada su propio número de RUT, como un entero antes del guión. Tome en cuenta que el ruteo debe contemplar las variables del programa, de la función **digito\_verificador** que usted escribió, y de la función **suma\_productos** suministrada.

```
rut = int(input('Ingrese su RUT: '))
d = digito_verificador(rut)
rut_completo = str(rut) + '-' + str(d)
print(rut_completo)
```

Descargue y modifique el [formato en Excel](#), y entregue el archivo modificado. Agregue las columnas necesarias, de acuerdo a las variables definidas en su función **digito\_verificador**. Si no tiene acceso a Excel, puede hacer el ruteo en papel u otra herramienta computacional, respetando el siguiente formato:

Programa			digito_verificador		suma_productos			
rut	d	rut_completo	rut	...	rut	suma_n	i	n

3. [50 %] Necesitamos un programa que determine el local y la mesa a la que deben concurrir a votar las personas. El local de votación se define a partir de la primera letra en los apellidos, como se muestra en la siguiente tabla. No importa si se ingresa 1 o 2 apellidos, se considera sólo la primera letra del *string*.

Primera letra de los apellidos	Local de votación
A – G	Peumo
H – O	Quillay
P – Z	Canelo

En el local Peumo votan quienes sus apellidos comienzan con una letra entre la A y la G. En el Quillay las personas cuyos apellidos comienzan con una letra entre la H y la O. Finalmente, en el Canelo votan quienes tienen apellidos que comienzan con una letra entre la P y la Z.

Una vez que se ha definido el local, se determina la mesa dentro del local. Para ello se considera la cantidad de letras en el nombre de la persona: se obtiene el resto de la división por 3 y luego se le suma 1. Esta operación entregará siempre una número de mesa entre 1 y 3.

Escriba un programa que lea el RUT (con guión y dígito verificador), los apellidos y el nombre de distintas personas, y determine e imprima el local y la mesa en que debe votar cada una de ellas. El programa debe terminar cuando se ingresa un RUT que no es válido, es decir, que su dígito verificador no corresponde.

Considere utilizar la función que escribió en la Pregunta 1. Puede suponer que la primera letra de los apellidos siempre aparecerá en mayúsculas.

Ejemplo:

```
Datos del votante:
RUT (con guión y dígito verificador): 6817253-5
Apellidos: Mistral
Nombre: Gabriela
Le toca votar en la mesa Quillay - 3
```

```
Datos del votante:
RUT (con guión y dígito verificador): 19489979-3
Apellidos: Bravo
Nombre: Claudio
Le toca votar en la mesa Peumo - 2
```

```
Datos del votante:
RUT (con guión y dígito verificador): 1-k
```