

Taller de Programación

Certamen 1

10 de Septiembre de 2019

Instrucciones:

- Lea atentamente el enunciado de cada uno de los problemas.
- Elija solo **TRES** de los CUATRO problemas del certamen.
- Para cada problema cree un archivo.py distinto. El nombre del archivo debe ser el número del problema (uno.py, dos.py, tres.py o cuatro.py)
- Comprima los problemas en un solo archivo **ZIP** y subalo a la sección Evaluación en <http://canvas.udd.cl>. Solo tiene una oportunidad para subir sus respuestas.
- Recuerde que usaremos un software de detección de plagio para detectar copia.

1. Entradas Lollapalooza (2 pts)

Escriba un programa que simule la compra de entradas para el Lollapalooza 2020, que se realizara durante el mes de marzo.

Para esto considere que el precio de la entrada depende de la fecha (mes) en que se compra la entrada y del tipo de entrada a comprar (pase diario o pase válido para los tres días que dura el evento) y que además se pueden incluir el servicio adicional de estacionamiento.

El programa debe recibir el mes en que se relizará la compra, el número de días que asistirá al evento, el número de entradas a comprar, preguntar si se desea incluir estacionamiento e imprimir el valor total de la compra.

Considere que los precios de las entradas de acuerdo a la fecha se distribuyen según la tabla adjunta y que el cargo por el servio extra de estacionamiento es \$5.000 por día.

	FECHA	Pase diario	Pase para 3 días
EARLY BIRD	Agosto		\$85.000
PREVENTA 1	Septiembre-Octubre		\$140.000
PREVENTA 2	Noviembre-Enero	\$85.000	\$180.000
ENTRADA GENERAL	Febrero-Marzo	\$100.000	\$250.000

Nota:

- Asuma que NO HAY un máximo de entradas disponibles por fecha de compra.

2. Serie (2 pts)

Escriba un programa que reciba un número x_0 a través del teclado e imprima una serie de números donde el valor del número x_{i+1} de la serie depende del valor del número anterior x_i según las siguientes reglas:

- Cuando x_i es **par**, el siguiente número de la serie es $x_i/2$
- Cuando x_i es **impar**, el siguiente número la serie es $3x_i + 1$

El programa debe parar cuando x_i sea igual a 1.

Ejemplo:

Si el número ingresado por teclado es 3 ($x_0 = 3$), el programa imprime `3, 10, 5, 16, 8, 4, 2, 1` .

3. El poder del número 7 (2 pts)

Programe el juego 'El poder del 7'. El objetivo de este juego es imprimir todos los números entre 1 y `n`, reemplazando con el string `aplauso` todos los números que contengan el dígito 7 o sean múltiplos de 7. Para el caso del número 77, este debe ser reemplazado por un aplauso doble `aplauso aplauso`.

Asuma que `n` es ≤ 100 y es ingresado por teclado. Su programa debe realizar las validaciones necesarias tal que `n` sea mayor a 1 y menor o igual a 100. Si no se cumplen estas reglas, su programa debe imprimir un mensaje de error `El número ingresado no es válido`.

Nota:

- Para saber si string `s1` esta contenido dentro de un string `s2` puede usar la instrucción `s1 in s2` que devuelve `True` cuando `s1` está en `s2` y `False` cuando no lo está. Por ejemplo, si `s1='1'` y `s2='123'`, `s1 in s2` retorna `True`.

4. Tweets (2 pts)

Escriba un programa que reciba un texto y palabras clave separadas por comas a través del teclado e imprima un **Tweet** siguiendo las siguientes reglas:

- Toda palabra del texto mayor a 4 letras debe ser acortada a las cuatro primeras 4 letras
- Las palabras acortadas deben reemplazar el ultimo caracter por el símbolo @
- Las palabras claves están contenida en el texto principal y son ingresadas separadas por coma.
- Las palabras claves deben ser convertidas a hashtags agregando el símbolo # al inicio cada palabra clave
- El tweet final no debe sobrepasar los 154 caracteres

Ejemplo:

```
##### Bienvenido al generador de Tweets #####
** Ingrese el mensaje: **
'No quiero ser alarmista pero mi padre tiene una empresa de alarmas y seguridad
y la voy a heredar.'
```

```
** Ingrese palabras clave: **
'alarmas,seguridad'
```

```
** Su texto transformado es: **
No qui@ ser ala@ per@ mi pad@ tie@ una emp@ de #al@ y #se@ y la voy a her@
```

Notas:

- La función `s.split('separador')` separa el string `s` y devuelve una lista de strings después de separar `s` por el separador. Por ejemplo si `s1= 'Me encanta programar'`, `s1.split()` devuelve `['Me', 'encanta', 'programar']` y si `s2= 'manzana, naranja, platano'`, `s2.split(',')` devuelve `['manzana', 'naranja', 'platano']`
- La función `' '.join(L)` convierte una lista `L` en un string donde cada elemento de la lista es separado por un espacio. Por ejemplo, si `L = ['1', '2', '3']`, `' '.join(L)` devuelve `'1 2 3'`.

