

UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA - UTEC

Tarea 2

Fecha de inicio: 07 de Febrero, 2021
Fecha de entrega: 13 de Febrero, 2021

Curso: Programación 1 (1100) – Entrega mediante: Gradescope

Indicaciones generales

1. Recuerda que la tarea es **individual**. Los casos de copia/plagio serán sancionados con nota cero (0) en la asignatura.
2. (a) La fecha límite de entrega es el **domingo 13 de Febrero a las 23:59 hrs.**
(b) Es altamente recomendable no esperar hasta la última hora.
(c) Gradescope desactivará automáticamente los envíos pasada dicha hora límite.
(d) **No se aceptarán entregas atrasadas ni entregadas por otros medios.**
3. Revisa bien lo que entregas, aunque en esta oportunidad podrás entregar ilimitadas veces la tarea, la última enviada será la evaluada.
4. Recuerda que Gradescope corrige automáticamente tu entrega. Dicha plataforma mostrará si has realizado correctamente las pruebas y mostrará algunos mensajes en color verde. Puedes ver un ejemplo de este caso en el anexo 1.
5. Es posible que hayas subido tu entrega pero hayas modificado algo que no se debió en el template. En ese escenario, Gradescope te mostrará algunos mensajes de error. Puedes ver un ejemplo de esto en el anexo 2.

Gradescope

1. Nosotros les proporcionaremos un código base de donde deberán partir para completar dicho ejercicio. Este archivo es llamado `solution.py` y lo encontrarán en la indicación de la tarea en CANVAS.
2. Al finalizar, **solo** subir el archivo `solution.py` (NO cambiar el nombre del archivo y NO comprimirlo).

3. Cada pregunta tiene diversos casos de prueba. Para obtener la nota completa en una pregunta, el algoritmo debe obtener la respuesta correcta en dichos casos de prueba.
4. Si un caso de prueba falla, visualizarán un mensaje de error con sugerencias. **Lee el error**, revisa el código e inténtalo de nuevo.
5. Los input de los casos de prueba son confidenciales.

Indicaciones específicas

1. En el anexo 1, se puede ver la plantilla de código.
2. Ustedes deben escribir dentro de la sección y a la misma altura de donde esta escrito *"Código comienza aquí"*. Además, no deben modificar nada debajo de *"Código acaba aquí"*. Recuerden tener cuidado con las indentaciones.
3. Los input del ejercicio se encuentran en la plantilla. Recuerden usar estas variables para resolver el ejercicio.
4. La respuesta del ejercicio debe ser impresa **específicamente** con *print()* para que Gradescope la tome en consideración. En el caso de funciones, esta debe ser retornada, según especifique la plantilla otorgada.
5. Al momento de la impresión de la respuesta, no adicionar texto. Imprimir **únicamente** el resultado que pide el ejercicio. De la misma manera con las funciones.

Organizando la data - (20 pts)

Le han solicitado que realice un programa que permita leer y transformar la información que registra un usuario. Para ello, deberá implementar una función que reciba un string, lo analice, y devuelva una lista con la información solicitada.

La función recibe una cadena que:

- Siempre empieza con un número.
- Después de un *'*'* siempre hay una palabra.
- Después de un *'-'* siempre hay otro número.
- Todos los caracteres que se encuentren entre *'#'* son irrelevantes.

Usted deberá leer el string, encontrar toda la información y retornarla en una lista.

Ejemplo 1.

```
1  cadena = '1889*902-Hola-Una frase'
```

```
1  [1889, 902, 'Hola', 'Una frase']
```

Ejemplo 2.

```
1  cadena = '002-Frase 1#caracteres 886 ppt#*1192-Fin'
```

```
1  [2, 'Frase 1', 1192, 'Fin']
```

Ejemplo 3.

```
1  cadena = '84-137-Mensaje*79#Holaaaa$$$'
```

```
1  [84, '137', 'Mensaje', 79]
```

1. Anexos

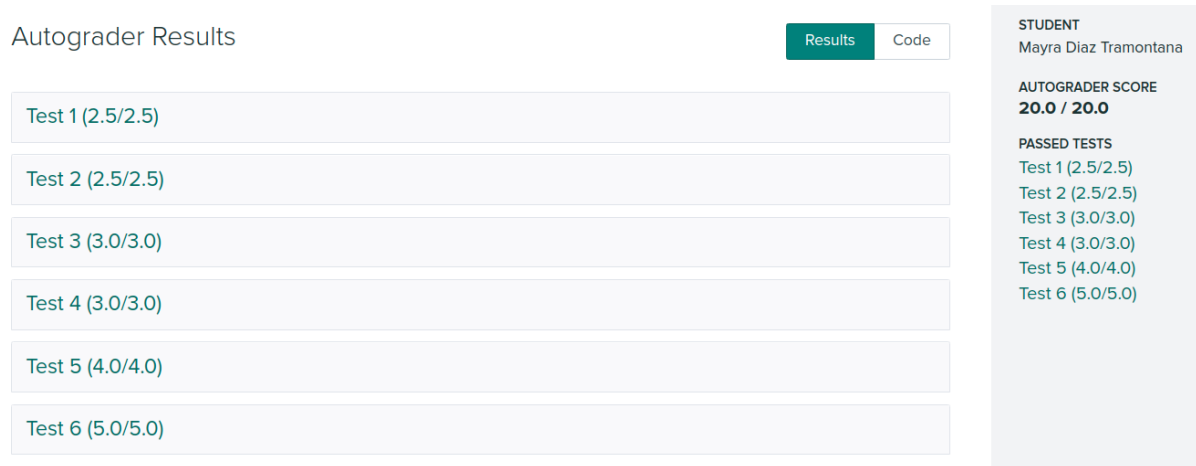


Figure 1: Casos de prueba correctos en Gradescope.

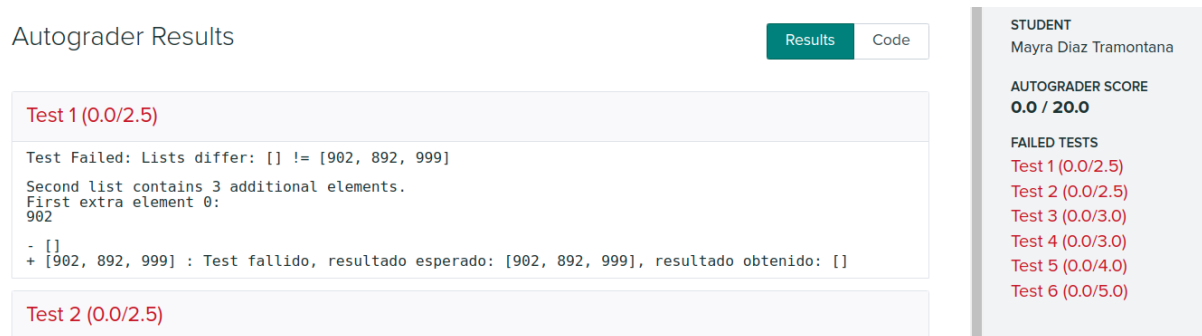


Figure 2: Entrega incorrecta en Gradescope.

```
1 class Solution():
2     def transformacion(self, cadena):
3         informacion = []
4         #SU SOLUCION EMPIEZA AQUI
5
6         #SU SOLUCION TERMINA AQUI
7         return informacion
```

Listing 1: Template solution.py.