

Tarea 1.1

Fecha de inicio: 31 de Enero, 2022
Fecha de entrega: 6 de Enero, 2022

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 6 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 ??.
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 ??

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 5 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 ??, se puede ver la plantilla de código del primer ejercicio de esta tarea.
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.
5. Al momento de la impresión de la respuesta, no adicionar texto. Imprimir **única-mente** el resultado del ejercicio.

Problema 1 - (5 pts)

¡Se acerca la semana universitaria! El equipo de marketing de UTEC ha pedido tu ayuda para encontrar, en su base de datos, todas las palabras especiales o "utecsinas" para poder utilizarlas en un nuevo comercial de televisión. Ellos te darán una palabra ha analizar y tú les responderás si es una palabra "utecsina" o no. Te han mandado un correo de apoyo que dice lo siguiente:

- Una palabra "utecsina" es aquella que incluye las letras **u, t, e** y **c**.
- Cada letra tiene que aparecer por lo menos **una** vez en la palabra y pueden estar en cualquier orden.
- Cada palabra ha analizar va a estar en minúsculas y tendrá una longitud mínima de 4 letras.
- Si la palabra es "utecsina" retornar un booleano True, en caso contrario, False.

Ejemplo 1.

Input :

```
1 palabra: cuenta
```

Output :

```
1 True
```

Explicación: La palabra "cuenta" es una palabra utecsina ya que tiene las cuatro letras **u, t, e** y **c**. cuenta

Ejemplo 2.

Input :

```
1 palabra: edificio
```

Output :

```
1 False
```

Explicación: La palabra "edificio" no es una palabra utecsina ya que solo tiene las letras e y c. edificio. Para que pueda ser una palabra utecsina le faltan las letras **t** y **u**.

Problema 2 - (10 pts)

Daniela está contigo en un proyecto grupal para el curso de Arte y tecnología. Ustedes deciden dibujar un rectángulo con Python para la presentación final. El profesor les avisó que, para tener buena nota, tienen que hacer un rectángulo que no sea tan sencillo. Para ello deciden separarse el trabajo, uno programa y el otro piensa como hacerlo especial. Al día siguiente, Daniela te muestra un rectángulo que tiene esta forma:

i\j	0	1	2	3	4
0					
1					
2					
3					

Ella ha decidido crear un rectángulo especial llenándolo de letras. Te explica que ha utilizado el par de coordenadas i, j para identificar cada celda del rectángulo, empezando desde 0,0. Para poder llenar cada celda se tienen 2 letras a elegir, una letra base y otra letra especial. Utilizando las reglas, se decide cuál letra va en cada celda dependiendo de su posición i, j . Al terminar su explicación, te da un papel con dichas reglas:

- Para empezar a construir el rectángulo necesitas estas 5 partes: longitud, altura, 1 letra base, 1 letra especial y 1 valor determinante.
- Longitud y altura **siempre** tienen un valor mayor que 0 y **nunca** tendrán el mismo valor.
- El valor determinante es igual al valor absoluto de la resta entre longitud y altura.
- La regla para definir qué letra va en cada posición es:
 - Si $i + j$ es múltiplo del valor determinante, entonces en esa posición i, j va una letra especial, en caso contrario, va una letra base.
- Se tiene que retornar un string que represente la forma del rectángulo. Para ello pueden concatenar un string letra por letra y crear un salto de línea concatenando el caracter especial "\n".

Ejemplo 1.**Input :**

```
1 longitud = 6
2 altura = 4
3 letra\_base = 'O'
4 letra\_especial = 'L'
```

Output :

```
1 LOLOLO\textbackslash nOLOLOL\textbackslash nOLOLOLO\textbackslash nOLOLOL
```

Explicación: El string "LOLOLO\nOLOLOL\nOLOLOLO\nOLOLOL" es equivalente a:

```
LOLOLO
OLOLOL
LOLOLO
OLOLOL
```

Pueden corroborar dicha respuesta imprimiendo el string.

$i \backslash j$	0	1	2	3	4	5
0	L	O	L	O	L	O
1	O	L	O	L	O	L
2	L	O	L	O	L	O
3	O	L	O	L	O	L

Antes de construir el rectángulo se tiene que calcular el valor determinante. $|\text{Longitud} - \text{altura}| \rightarrow |6 - 4| = |-2| = 2$. El valor determinante es 2.

Por cada celda i, j se hace la pregunta si $i + j$ es múltiplo del valor determinante; si cumple, en esa celda iría una L , en caso contrario una O .

Por ejemplo, se aprecia que en la posición 1, 1 está la letra especial L . Ello ocurre ya que cumple la regla: $i + j = 2$ y 2 es múltiplo de 2.

Otro ejemplo, en la posición 3, 2 está la letra base O . Ello ocurre ya que no cumple la regla: $i + j = 5$ y 5 no es múltiplo de 2.

1. Anexos

Autograder Results

Results

Code

Autograder Output (hidden from students)

```
True
LOLOLO
OLOLOL
LOLOLO
OLOLOL
```

Pregunta 1: Test 1 (1.0/1.0)

Pregunta 1: Test 2 (1.0/1.0)

Pregunta 1: Test 3 (1.0/1.0)

Pregunta 1: Test 4 (1.0/1.0)

Pregunta 1: Test 5 (1.0/1.0)

STUDENT

Mayra Diaz Tramontana

AUTOGRADER SCORE

15.0 / 15.0

PASSED TESTS

Pregunta 1: Test 1 (1.0/1.0)
Pregunta 1: Test 2 (1.0/1.0)
Pregunta 1: Test 3 (1.0/1.0)
Pregunta 1: Test 4 (1.0/1.0)
Pregunta 1: Test 5 (1.0/1.0)
Pregunta 2: Test 1 (2.0/2.0)
Pregunta 2: Test 2 (2.0/2.0)
Pregunta 2: Test 3 (2.0/2.0)
Pregunta 2: Test 4 (2.0/2.0)
Pregunta 2: Test 5 (2.0/2.0)

Figure 1: Casos de prueba correctos en Gradescope.

Autograder Results

Results

Code

Autograder Output (hidden from students)

```
None
None
```

Pregunta 1: Test 1 (0.0/1.0)

Test Failed: None != True : Test 1 fallido. Resultado esperado: True. Resultado obtenido: None.

Pregunta 1: Test 2 (0.0/1.0)

Test Failed: None != False : Test 2 fallido. Resultado esperado: False. Resultado obtenido: None.

Pregunta 1: Test 3 (0.0/1.0)

Test Failed: None != True : Test 3 fallido. Resultado esperado: True. Resultado obtenido: None.

STUDENT

Mayra Diaz Tramontana

AUTOGRADER SCORE

0.0 / 15.0

FAILED TESTS

Pregunta 1: Test 1 (0.0/1.0)
Pregunta 1: Test 2 (0.0/1.0)
Pregunta 1: Test 3 (0.0/1.0)
Pregunta 1: Test 4 (0.0/1.0)
Pregunta 1: Test 5 (0.0/1.0)
Pregunta 2: Test 1 (0.0/2.0)
Pregunta 2: Test 2 (0.0/2.0)
Pregunta 2: Test 3 (0.0/2.0)
Pregunta 2: Test 4 (0.0/2.0)
Pregunta 2: Test 5 (0.0/2.0)

Figure 2: Entrega incorrecta en Gradescope.

```
1 class Solution:
2     # RECORDAR RETORNAR LA RESPUESTA
3
4     def palabrasUtecsinas(self, palabra):
5         #TU ALGORITMO VA AC
6         return
7
8     def rectanguloEspecial(self, longitud, altura, letra_base,
9        letra_especial):
10        #TU ALGORITMO VA AC
11        return
12
13 # === CASOS EJEMPLO ===
14 print(Solution().palabrasUtecsinas("cuenta")) # output en consola: True
15 print(Solution().rectanguloEspecial(6,4,"O","L"))
16 """
17 output en consola:      LOLOLO
18                      OLOLOL
19                      LOLOLO
20                      OLOLOL
21 """
```

Listing 1: Template solution.py.