

#### Indicaciones específicas:

- Esta evaluación contiene 10 páginas (incluyendo esta página) con 4 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta y tu código de estudiante. Por ejemplo:
  1. p1\_2020010202.py
  2. p2\_2020010202.py
  3. p3\_2020010202.py
  4. p4\_2020010202.py
- Luego deberás incluir estos archivos en una carpeta con nombre **pc2**; para que finalmente envíes esta carpeta comprimida **pc2.zip** a [www.gradescope.com](http://www.gradescope.com)

#### Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
  - Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- Para los alumnos de las carreras de Ingeniería
  - Capacidad de aplicar conocimientos de ingeniería (**nivel 2**).

## Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	5	
2	5	
3	5	
4	5	
Total:	20	

1. (5 points) CAPTCHA es una abreviatura que viene de las siglas de "Completely Automated Public Turing test to tell Computers and Humans Apart", que literalmente significa Test Público Completamente Automatizado de Turing para Diferenciar a los Seres Humanos de las Computadoras. En ese sentido se pide realizar un algoritmo para generar dicho conjunto de caracteres de acuerdo a las siguientes reglas.

- Que el primero y el último carácter sea 1 letra en Mayúscula [A-Z]
- Que tenga 2 números iguales [0-9]
- Que el resto de caracteres sean minúsculas. [a-z]

Tener en cuenta que se requiere la creación explícita de la función:

Listing 1: Código base

```
letras = "abcdefghijklmnopqrstuvwxyz"

def generarToken():
    token = ""
    # Escribe tu código aquí

    return token

print(generarToken())
```

Nota: Debe completar su algoritmo en la función *generarToken* y solo debe usar las letras consideradas en la variable *letras*.

Algunos ejemplos de diálogo de este programa serían:

Listing 2: Ejemplo 1

```
Captcha: Q9f9qT
```

Listing 3: Ejemplo 2

```
Captcha: X2y6lV
```

Listing 4: Ejemplo 3

```
Captcha: M0n1bR
```

Listing 5: Ejemplo 4

```
Captcha: T4s2oQ
```

La rúbrica para esta pregunta es:

Criterio	Logrado	Parcialmente Logrado	No Logrado
Algoritmo	Es preciso, finito y hace exactamente lo que el enunciado requiere (3pts)	Es preciso, finito y hace la mitad o más de lo que el enunciado requiere (1pts)	Hace menos de la mitad de lo que el enunciado requiere (0pts).
Sintáxis	Todas las sentencias son correctas y no hay errores de sintáxis. (1pts).	Mas de la mitad de las sentencias son correctas y no hay errores de sintáxis. (0.5pts).	Menos de la mitad de las sentencias son correctas (0pts).
Legible	El algoritmo es correcto y el nombre de todas las variables y funciones son descriptivas (1pts)	El algoritmo es correcto y el nombre de la mitad de las variables y funciones son descriptivas (0.5 pts)	El algoritmo es correcto y el nombre de menos la mitad de las variables y funciones son descriptivas (0 pts).

2. (5 points) Implementar una función que reciba como parámetro una lista y proceda a eliminar el segundo elemento y traslade el penúltimo elemento en su reemplazo.

Algunos ejemplos de diálogo de este programa serían:

Listing 6: Ejemplo 2

```
input: [14,15,25,35,14,48,86,48,14,16,80]
output: 14, 16, 25, 35, 14, 48, 86, 48, 14, 80
```

Listing 7: Ejemplo 1

```
input: ["Juan", "Luis", "Pedro", "Maria", "Carlos", "Pedro", "
      Manuel", "Miriam", "Jorge"]
output: 'Juan', 'Miriam', 'Pedro', 'Maria', 'Carlos', 'Pedro
      ', 'Manuel', 'Jorge'
```

Listing 8: Ejemplo 2

```
input: ["C","A","E","B","D","F","G","H","J","K","A"]
output: 'C', 'K', 'E', 'B', 'D', 'F', 'G', 'H', 'J', 'A'
```

La rúbrica para esta pregunta es:

Criterio	Logrado	Parcialmente Logrado	No Logrado
Algoritmo	Es preciso, finito y hace exactamente lo que el enunciado requiere (2pts)	Es preciso, finito y hace la mitad o más de lo que el enunciado requiere (1pts)	Hace menos de la mitad de lo que el enunciado requiere (0pts).
Funciones	Hace uso correcto de las funciones con sus parametros (1pts)	Uso parcialmente de funciones (0.5pts)	no hace uso correcto de las funciones (0pts).
Sintáxis	Todas las sentencias son correctas y no hay errores de sintáxis. (1pts).	Mas de la mitad de las sentencias son correctas y no hay errores de sintáxis. (0.5pts).	Menos de la mitad de las sentencias son correctas (0pts).
Legible	El algoritmo es correcto y el nombre de todas las variables y funciones son descriptivas (1pts)	El algoritmo es correcto y el nombre de la mitad de las variables y funciones son descriptivas (0.5 pts)	El algoritmo es correcto y el nombre de menos la mitad de las variables y funciones son descriptivas (0 pts).

3. (5 points) El servicio de taxi por aplicación móvil tales como Uber, Cabify, EasyTaxi, TaxiBeat aplicaciones muy populares y es de gran utilidad, sin embargo es necesario conocer los factores que toman en cuenta al momento de establecer sus tarifas por cada servicio.
- El sistema de tarifas de Uber es bastante sencillo. Cobran 1,20 soles por cada kilómetro recorrido y 0,10 soles adicionales por cada minuto en marcha. Es decir, que si que hacemos cuatro kilómetros y tardamos 10 minutos, la carrera nos cuesta 5,80 soles.
  - Las tarifas de Cabify son incluso más sencillas que las de Uber, ya que sólo cobra una cantidad por kilómetro recorrido (1,65 soles para los 20 primeros, 1,10 para el resto), independientemente del tiempo.
  - Aunque desde hace unos años las tarifas de Taxi Beat han ido reduciendo su complejidad, siguen siendo aun confusos para el usuario final. En esencia hay que tener en cuenta tres elementos: una cantidad fija por inicio del servicio (2,40 soles), una cantidad por kilómetro (1,05 soles) y una cantidad por tiempo transcurrido (0,34 soles por minuto).

Implementar una funcion *precio* que reciba como parametro el nombre del servicio, distancia y el tiempo que dura una carrera. la función debe devolver el precio a pagar en cada servicio.

Listing 9: Código base

```
def precio(servicio, distancia, tiempo):
    precio = 0
    # Escribe tu codigo aqui

    return precio

servicios = ["UBER", "Cabify", "TaxiBeat"]
distancia = "_" #valor de la distancia
tiempo = "_" #valor del tiempo

for i in servicios:
    print(i, ":", round(precio(i, distancia, tiempo), 2))
```

Algunos ejemplos de diálogo de este programa serían:

Listing 10: Ejemplo 1

```
distancia = 50
tiempo = 20
-----Output-----
UBER : 62.0
Cabify : 66.0
TaxiBeat : 61.7
```

Listing 11: Ejemplo 1

```

distancia = 20
tiempo = 15
-----Output-----
UBER : 25.5
Cabify : 33.0
TaxiBeat : 28.5

```

Listing 12: Ejemplo 1

```

distancia = 200
tiempo = 160
-----Output-----
UBER : 256.0
Cabify : 231.0
TaxiBeat : 266.8

```

La rúbrica para esta pregunta es:

Criterio	Logrado	Parcialmente Logrado	No Logrado
Algoritmo	Es preciso, finito y hace exactamente lo que el enunciado requiere (3pts)	Es preciso, finito y hace la mitad o más de lo que el enunciado requiere (1pts)	Hace menos de la mitad de lo que el enunciado requiere (0pts).
Sintáxis	Todas las sentencias son correctas y no hay errores de sintáxis. (1pts).	Mas de la mitad de las sentencias son correctas y no hay errores de sintáxis. (0.5pts).	Menos de la mitad de las sentencias son correctas (0pts).
Legible	El algoritmo es correcto y el nombre de todas las variables y funciones son descriptivas (1pts)	El algoritmo es correcto y el nombre de la mitad de las variables y funciones son descriptivas (0.5 pts)	El algoritmo es correcto y el nombre de menos la mitad de las variables y funciones son descriptivas (0 pts).

4. (5 points) Alessia es una niña muy afortunada y se ganó un nuevo juguete **“Don’t Break the Ice”**, que es un juego de mesa y se juega con un conjunto de “bloques de hielo” de plástico (generalmente 6x6), un mazo en miniatura y un soporte. Un bloque de hielo es más grande que el resto (Normalmente 2x2), y ya sea un hombre, un oso polar o un Pingüino, se encuentra en este bloque. Los jugadores se turnan para quitar bloques tocando con el mazo. El juego termina cuando un jugador que “rompe el hielo”, lo que provoca que el hombre, el oso o Pingüino se caiga. El jugador que haya eliminado la mayoría de los bloques sin “romper el hielo” es el ganador.



Figure 1: Juego “Don’t Break the Ice”

Alessia, decide experimentar con su nuevo juguete y extrañamente cuando golpea el bloque grande los demás bloques se caen, pero, cuando golpea un bloque pequeño se cae solo el bloque golpeado.

Desarrolle un algoritmo que permita modelar las ocurrencias de Alessia. Para ello, considere utilizar una matriz dinámica de 6x6, además, cada bloque pequeño estará representado por cualquier número mayor a 2 o menor a 10, el bloque grande estará representado por 1s, los bloques caídos serán representados con 0.

### Input

- Se tiene la matriz 6x6 que estan representado con números enteros como se muestra a continuación.

3	3	6	7	8	8
5	7	8	1	1	3
6	9	5	1	1	8
7	6	9	3	8	8
3	3	6	7	7	8
3	3	6	7	8	3

### Ouput

- Se espera Imprimir una nueva matriz con los bloques caídos y el número del bloque golpeado.

Algunos ejemplos de este programa serían:



Listing 13: Ejemplo 1

```
Ingrese ubicacion del bloque: 0 2
Output:
3 3 0 7 8 8
5 7 8 1 1 3
6 9 5 1 1 8
7 6 9 3 8 8
3 3 6 7 7 8
3 3 6 7 8 3
```

Listing 14: Ejemplo 2

```
Ingrese ubicacion del bloque: 1 4
Output:
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

Listing 15: Ejemplo 3

```
Ingrese ubicacion del bloque: 2 4
Output:
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

La rúbrica para esta pregunta es:

Criterio	Logrado	Parcialmente Logrado	No Logrado
Algoritmo	Es preciso, finito y hace exactamente lo que el enunciado requiere (2pts)	Es preciso, finito y hace la mitad o más de lo que el enunciado requiere (1pts)	Hace menos de la mitad de lo que el enunciado requiere (0pts).
Sintáxis	Todas las sentencias son correctas y no hay errores de sintáxis. (1pts).	Mas de la mitad de las sentencias son correctas y no hay errores de sintáxis. (0.5pts).	Menos de la mitad de las sentencias son correctas (0pts).
Legible	El algoritmo es correcto y el nombre de todas las variables y funciones son descriptivas (1pts)	El algoritmo es correcto y el nombre de la mitad de las variables y funciones son descriptivas (0.5 pts)	El algoritmo es correcto y el nombre de menos la mitad de las variables y funciones son descriptivas (0 pts).
Iteración	Recorre adecuadamente los elementos de la matriz (1pts)		No recorre los elementos de la matriz de forma programática y solo funciona en algunos casos. (0 pts).