 PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA Facultad de Ingeniería y Ciencias Agrarias	<b>EXÁMEN PRÁCTICO</b>	<b>23 / 02 / 2021</b>	<b>FINAL</b>	
		<b>INFORMÁTICA GENERAL</b>		
	<i>En máquina, virtual presencial y a distancia</i>		<b>TEMA 1</b>	

▬ INTRODUCCIÓN

Exam.net nos ofrece, el siguiente menú:

- 1. **Área de escritura:** Es un editor de texto que lo podés usar como borrador, anotador, anotar comentarios del enunciado, y registrar las aclaraciones que realicen los docentes de manera oral durante el examen, etc.
- 2. **>\_ Programación:** Es una herramienta de programación. Podés escribir el código en lenguaje de programación y probarlo. Debes seleccionar el lenguaje Python.
- 3. **EXAMEN ...:** Aquí vas a encontrar el texto con la consigna del examen.
- 4. **Chat de profesores:** Podrás realizar preguntas a tus profesores utilizando este chat.
- 5. **Enviar examen:** Es el link que te lleva al envío del examen, donde estarías realizando la entrega de tu examen. *En síntesis..terminaste el examen !!!*

**El examen práctico contiene DOS ejercicios a desarrollar. IMPORTANTE LEER:**  
El estudiante deberá guardar **un** archivo por **cada** ejercicio. Grabar con el nombre del ejercicio (**EJ01** o **EJ02** ) según corresponda.

Además, para identificar el ejercicio es obligatorio escribir dentro del archivo codificado (como comentario en los dos primera línea), el número de ejercicio que están resolviendo y el apellido y nombre del estudiante que lo resuelve. Por ejemplo para el ejercicio 1 (**EJ01**), para la estudiante **Suarez Alejandra** deberá quedar :


```
### EJ01 ###
### SUAREZ ALEJANDRA ###
```

Por favor, identificar todos lo ejercicios de esta manera

Es importante que si grabas varias versiones de un mismo ejercicio, antes de enviar el examen, elimina las versiones que están de más, y deja una sólo por ejercicio (la última). De esta manera recibiremos un archivo por ejercicio.  
**No deben usar input ni tampoco archivos la resolución de ninguno de los ejercicios.**

**IMPORTANTE: Sólo podemos usar los siguientes elementos del lenguaje para resolver los problemas en los **FINALES** o evaluaciones en la cátedra. Y sujeto a indicaciones del profesor.**

- DELIMITADORES: # ' " , : . \ ( ) [ ] { }
- OPERADORES: + - \* \*\* / // < > <= >= == != = += -=
- PALABRAS RESERVADAS: and def elif else False for if import in None not or return True while
- FUNCIONES INTEGRADAS: bool() chr() dict() float() id() input() int() len() list() max() min() open() ord() print() range() round() str() tuple() type()
- FUNCIONES LIBRERÍA math: math.pow() math.sqrt() - FUNCIONES LIBRERÍA random: random.randint()
- MÉTODOS DE STRING: .format() .split()
- MÉTODOS DE LIST: .append() .insert() .pop() .remove()
- MÉTODOS DE DICT: .get() .keys() .pop() .values()
- MÉTODOS DE FILE: .close() .readline() .readlines() .write()

 PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA Facultad de Ingeniería y Ciencias Agrarias	<b>EXÁMEN PRÁCTICO</b>		23 / 02 / 2021	FINAL	
			INFORMÁTICA GENERAL		
	En máquina, virtual presencial y a distancia			TEMA 1	

DESCRIPCIÓN DE LOS DATOS DEL PROBLEMA A RESOLVER

“Un parque eólico es un conjunto de molinos de viento generando energía eléctrica”. - Nos encargan el análisis de los datos de producción de energía del sistema del parque eólicos de un país. Para ello se cuenta con información en dos archivos (- No se sabe la cantidad de registros que hay en cada archivo -). A continuación el detalle de los archivos:

	energia.csv	parque.csv	
Estructura de datos	AMD [string] ID_parque [int] cantidad [float]  (AMD: Ver detalle del formato más abajo.)	ID_parque [int] nombre [string] cantidadMolinos [int]	
Ejemplo de Contenido	101205, 1, 24.2 110607, 8, 54.4 120318, 3, 18.1 090501, 9, 88.4 101209, 1, 26.8 101217, 3, 22.4 190101, 8, 44.0	1,Rosario, 8 6,San Martin, 4 8,Lavalle, 10 3,Esperanza, 3 7,General Pico, 9 9,P.Madryn, 4	

**Formato: AMD (Año Mes Día).** Dentro de un string de seis caracteres numéricos hay una fecha representada según se indica: **AAMMDD**. Ejemplo: 120318. El año (**AA**) es 12, el mes (**MM**) es 03, el día (**DD**) es 18.

- **energia.csv**: Contiene tres campos. - **AMD**: Fecha en que se realizó la medición de la energía entregada por el parque (ver detalle del formato). - **ID\_parque**: El identificador del parque eólico. - **cantidad**: La cantidad de energía entregada por el parque. *Son claves combinadas los campos {AMD,ID\_parque}; es decir la combinación de los valores de dichos campos son única , no se repiten en el archivo (subrayada en la estructura de datos).*
- **parque.csv**: Contiene tres campos. - **ID\_parque**: Identificador único e irrepetible para el parque. - **nombre**: El nombre correspondiente al parque.- **cantidadMolinos**: Cantidad de molinos eólicos que tiene el parque. *Es clave el campo {ID\_parque}; es decir el valor de dicho campo es único , no se repite en el archivo (subrayada en la estructura de datos).*


**Se cuenta con dos variables del tipo lista lstEnergia y lstParque.** Cada lista tiene el contenido proveniente de la lectura completa del archivo .csv correspondiente, utilizando el método `.readlines()`. Los campos del archivo están separados por coma (,) y no está ordenado por ningún campo.

**EJEMPLO DE LISTAS** - Obtenidas a partir de la lectura de cada archivo  
Resulta de un `readlines()` a **energia.csv**

```
lstEnergia = ['101205, 1, 24.2\n', '110607, 8, 54.4\n', '120318, 3, 18.1\n', '090501, 9, 88.4\n', '101209, 1, 26.8\n', '101217, 3, 22.4\n', '190101, 8, 44.0\n']
```

Resulta de un `readlines()` a **parque.csv**

```
lstParque = ['1,Rosario, 8\n', '6,San Martin, 4\n', '8,Lavalle, 10\n', '3,Esperanza, 3\n', '7,General Pico, 9\n', '9,P.Madryn, 4\n']
```

 <b>UCA</b> <small>PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA</small> <b>Facultad de Ingeniería y Ciencias Agrarias</b>	<b>EXÁMEN PRÁCTICO</b>	<b>23 / 02 / 2021</b>	<b>FINAL</b>	
		<b>INFORMÁTICA GENERAL</b>		
	<i>En máquina, virtual presencial y a distancia</i>		<b>TEMA 1</b>	

EJ01

Desarrollar la función `minimos(lstEnergia)` que a partir de la información que contiene `lstEnergia` la función deberá determinar:

- El año “AA” donde se produce la menor energía, y la cantidad de energía generada “cEnergiaA” en dicho año.
- El mes (con el año) “AAMM” donde se produce la menor energía, y la cantidad de energía generada “cEnergiaM” para dicho mes / Año.

La función deberá retornar una lista conteniendo dos tuplas, donde cada tupla contiene los valores determinadas anteriormente, cómo se muestra a continuación:

Formato de la lista a retornar ->

[ (AA, cEnergiaA) , (AAMM, cEnergiaM) ]

Este es un main de ejemplo el cual sugerimos utilizar para ejecutar pruebas con la función del ejercicio:

```
def main():
    print("Prueba para el EJ01")
    lstEnergia = [ '101205,1,24.2\n', '110607,8,54.4\n', '120318,3,18.1\n',
'090501,9,88.4\n', '101209,1,26.8\n', '101217,3,22.4\n', '190101,8,44.0\n' ]
    print(minimos(lstEnergia))
```

Al ejecutar el main planteado arriba, la SALIDA ESPERADA es:

```
>>> Prueba para el EJ01

[('12', 18.1), ('1203', 18.1)]
```


EJ02

Desarrollar la función `energiaTotal(n, lstEnergia, lstParque)`, que recibe por parámetro un valor entero `n` (que representa una cantidad) y las dos listas con los datos. A partir de la información que contiene `lstEnergia` y `lstParque`, determinar:

- Los `n` parques eólicos que **MAYOR** Cantidad de **Energía** total es producida por el **Parque**(CEP). (*nota1*). Y para cada uno de estos en `parques`, determinar (CEM): la **Cantidad de Energía** que produce cada **Molino** de un parque determinado (*nota2*).
  - (*nota1*): (CEP) se obtiene de sumar toda la energía producida de cada día, para un determinado parque.
  - (*nota2*): (CEM) Se obtiene de dividir la cantidad total de energía que produce un parque dividido la cantidad de molino que tiene dicho parque.

La función deberá retornar una lista de lista de `n` elementos donde cada elemento de la sublista contiene el nombre del parque, el CEP y el CEM para dicho parque.

Estructura de cada elemento de la sublista a retornar ->

 PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA Facultad de Ingeniería y Ciencias Agrarias	<b>EXÁMEN PRÁCTICO</b>		23 / 02 / 2021	FINAL	
			INFORMÁTICA GENERAL		
		En máquina, virtual presencial y a distancia		TEMA 1	

[ nombre\_parque, CEP , CEM ]

Este es un main de ejemplo el cual sugerimos utilizar para ejecutar pruebas con la función del ejercicio:

```
def main():
    print("Prueba para el EJ02")
    lstEnergia = ['101205,1,24.2\n', '110607,8,54.4\n', '120318,3,18.1\n',
'090501,9,88.4\n', '101209,1,26.8\n', '101217,3,22.4\n', '190101,8,44.0\n']
    lstParque = ['1,Rosario,8\n', '6,San Martin,4\n', '8,Lavalle,10\n',
'3,Esperanza,3\n', '7,General Pico,9\n', '9,P.Madryn,4\n']
    print(energiaTotal(3, lstEnergia, lstParque))
```

Al ejecutar el main planteado arriba, la SALIDA ESPERADA es:

```
>>> Prueba para el EJ02

[['Lavalle', 98.4, 9.84], ['P.Madryn', 88.4, 22.1], ['Rosario', 51.0, 6.375]]
```

FIN