Exam.

Cerrar sesión Ayuda

EJ T1

```
def comparar(a,b,c):
    if((a and (b or c)) != ((a and b)or(a and c))):
        return True
    else:
        return False
```

¿Qué retorna la función?

a) siempre retorna False

b) siempre retorna True

c) El valor de retorno depende de los valores de las variables que se pasan por parámetro

d) No retorna nada, pues hay error de sintaxis

h) Ninguna de las anteriores

EJ T2

```
def f2(x):
    x=x+10
    return x

def f1():
    x=10
    print(str(f2(x))+"-",end="")

x=5
f1()
print ("{0:d}".format(f2(x)))
```

¿Qué imprime al ejecutar el código anterior?

d) No retorna nada, pues hay error de sintaxis

h) Ninguna de las anteriores

EJ T2

```
def f2(x):
    x=x+10
    return x/2

def f1():
    x=10
    print(str(f2(x))+"-",end="")

x=5
f1()
print ("{0:d}".format(f2(x)))
```

¿Qué imprime al ejecutar el código anterior?

a) 20.15-20

b) 15-20-15

c) 15-20

d) 20-15

e) Ninguna de las anteriores

EJ T3

Completar las líneas en blanco (_____) , sin agregar ni cambiar renglones al código mostrado, de la función **esPalindromo** para que retorne `True` si la palabra pasado por parámetro es un palíndromo o retorne `False` en caso contrario. Asuma que siempre se recibe una sola palabra y está en minúsculas. *Palindromo es una palabra o expresión que es igual si se lee de izquierda a derecha que de derecha a izquierda.* Ej's: Anana, ana, rayar, ...etc.

```
def esPalindromo (palabra):
    palindromo=True
    for i in range(_____):
        if(palabra[_____] !=palabra[ i ]):
            _____=False
    return _____
```

EJ T4

Completar las líneas en blanco (_____), sin agregar ni cambiar renglones al código mostrado de la función `archivoAListaDeRenglones` para que la función abra y lea el archivo cuyo nombre se recibe por parámetro, para crear y retornar una lista de strings donde cada string se corresponde con un renglón del archivo leído.

```
def archivoAListaDeRenglones(nombreArchivo):
    arch= _____
    lista= _____
    _____
    return _____
```

UCA
Facultad de Ingeniería y Ciencias Agrarias

EXAMEN PRÁCTICO | 04 / 08 / 2020 | FINAL

INFORMÁTICA GENERAL

En máquina, virtual, presencial y a distancia | TEMA 1

EJ01

Desarrollar la función `contagiados(fecha, lst_ciudad, lst_contagios)` que recibe por parámetro las listas y una `fecha` (en Formato AMD). La función deberá retornar una lista con la cantidad total de contagiados (*diagnosticados positivos*) por cada ciudad, para la fecha (*fecha de contagio*) pasada por parámetro. Cada elemento de la lista a retornar deberá ser una tupla de dos elementos. Estos dos elementos deben ser: *<el nombre de la ciudad>* y *<la cantidad de contagiados para dicha ciudad> para la fecha pasada por parámetro*. La lista a retornar debe contemplar los casos de TODAS las ciudades (incluso las que no presenten ningún positivo).

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")
    lst_ciudad = ['223,Parana\n', '114,Merlo\n', '218,Guaymallen\n', '132,C. Rivadavia\n',
'341,Adolfo Alsina\n', '484,Jose C. Paz\n']
    lst_contagios = ['10,33,114,200518\n', '11,31,223,200519\n', '12,27,218,200319\n',
'13,26,132,200516\n', '14,74,341,200319\n', '15,62,484,200606\n', '16,46,218,200709\n',
'17,55,132,200630\n', '18,55,341,200612\n', '19,54,484,200701\n', '1,23,114,200315\n',
'2,55,223,200519\n', '3,34,218,200319\n', '4,33,132,200425\n', '5,27,341,200422\n',
'6,21,484,200501\n', '7,31,114,200503\n', '8,44,114,200513\n', '9,44,218,200519\n']

    print(contagiados(200519,lst_ciudad,lst_contagios))
```

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

```
>>> Prueba para el EJ01
[('Parana', 2), ('Merlo', 0), ('Guaymallen', 1), ('C. Rivadavia', 0), ('Adolfo Alsina', 0), ('Jose C. Paz', 0)]
```

EJ02

Desarrollar la función `nMaxContagios(n, lst_ciudad, lst_contagios)` que recibe por parámetro las listas y un tercer parámetro `n` que es un valor numérico entero que representa una

PRÁCTICO

.21.FINAL.JUL.B.PRACTICA

Se actualiz

[redacted]

EJ01

Desarrollar una función llamada **mesGeneracion(***lsArch,cant***)** que reciba 2 parámetros. `lsArch` es una lista que contiene el contenido completo del archivo (cómo se indica arriba). `cant` qué es un entero que representa un cantidad

La función **mesGeneracion** deberá imprimir un listado con los `cant` meses de mayor generación de energía. El listado debe estar ordenado de mayor a menor por valor de generación de energía. Para obtener la generación para un mes se deberá sumar todas las generaciones para dicho mes para cualquier año. (*Ver el ejemplo de salida para mayor ejemplificación*)

Aclaraciones EJ 01: *Los profesores podrán probar la función* **mesGeneracion** *con distintos valores de* `cant` *y/o otros contenidos de* `lsArch`

AYUDA: Ejemplos de uso del .format

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

```
def main():
    lsArch = ['central,region,tecnologia,fuente_generacion,generacion_neta_MWh,anio_mes\n',
'CAPE,COMAHUE,TURBO GAS,Termica,21668984,2020-01\n', 'AESP,BUENOS AIRES,TURBO
VAPOR,Termica,193938412,2012-01\n', 'ALEM,NORESTE,MOTOR DIESEL,Termica,4882979,2017-07\n',
'ALICHI,COMAHUE,TURB HIDRAULICA,Hidraulica,1002141172,2018-07\n', 'APEGHI,PATAGONICA,TURB
HIDRAULICA,Hidraulica,11705,2018-08\n', 'ARAT,NORDESTE,MOTOR DIESEL,Termica,7670154,2018-07\n',
'APAR,BUENOS AIRES,TURBO VAPOR,Termica,1077474,2015-09\n',
'ARAUCO,NOROESTE,EOLICA,Renovable,844759,2017-02\n', 'ARROMI,COMAHUE,TURB
HIDRAULICA,Hidraulica,2961,2012-01\n', 'ATUC,BUENOS AIRES,NUCLEAR,Nuclear,217003,2019-03\n',
'ULN1FV,CUYO,FOTOVOLTAICA,Renovable,5779061,2020-03\n', 'VGES,BUENOS AIRES,TURBO
GAS,Termica,43969,2020-02\n', 'VLOMEO,BUENOS AIRES,EOLICA,Renovable,1171871,2020-05\n']

    print("\nCASO PRUEBA 01 - cant=5")
    mesGeneracion (lsArch,5)
    print("\nCASO PRUEBA 02 - cant=1")
    mesGeneracion (lsArch,1)
    print("\nCASO PRUEBA 03 - cant=100")
    mesGeneracion (lsArch,100)
    print("\nCASO PRUEBA 04 - cant=0")
    mesGeneracion (lsArch,0)
main()
```

SALIDA ESPERADA Al ejecutar el main planteado arriba

```
CASO PRUEBA 01 - cant=5
MES     GENERACION
01      215810357
07      107911326
03      10099063
05      1171872
09      1077474

CASO PRUEBA 02 - cant=1
MES     GENERACION
01      215810357

CASO PRUEBA 03 - cant=100
MES     GENERACION
01      215810357
07      107911326
03      10099063
05      1171872
09      1077474
02      888728
08      11705

CASO PRUEBA 04 - cant=0
MES     GENERACION
```

EJ02

Desarrollar la función **archAlst**(lsArch,col,ord) que recibe 3 parámetros. `lsArch` es una lista que contiene el contenedo completo del archivo (*cómo se indica arriba*). `col` es un entero que indicará un número de columna y `ord` es un número entero, si `ord` es igual a `0` indica orden 'de *menor a mayor* (ascendente) si `ord` es *distinto* de `0` indica orden 'de *mayor a menor* (descendente)

La función **archAlst** deberá retornar una lista de lista con el contenido del archivo pero con la siguiente característica

- Cada sublista es un renglón del archivo y cada elemento de la sublista es un campo del renglón correspondiente
- La lista debe estar ordenada por el número de columna `col` pasado por parámetro, y de forma ascendente o descendente según indica `ord` también pasado por parámetro
    - Si `col` está fuera de rango de columnas válidas, entonces la función deberá retornar la lista de lista con el orden original, es decir tal cual vienen los datos en `lsArch`
- Observar que el archivo contiene un renglón con los títulos de las columnas, este renglón no debe intervenir en el ordenamiento, siempre deben quedar primero de todos en la lista (es decir en la posición cero)

Aclaraciones EJ_02 Los profesores podrán probar la función **archAlst** con distintos valores de `col` y de `ord`, y/o con otros contenidos de `lsArch`.

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

```
def main():
    lsArch = ['central,region,tecnologia,fuente_generacion,generacion_neta_MWh,anio_mes\n',
'CAPE,COMAHUE,TURBO GAS,Termica,21868984,2020-07\n', 'AESP,BUENOS AIRES,TURBO
'CAPE,COMAHUE,TURBO GAS,Termica,21868984,2020-07\n', 'AESP,BUENOS AIRES,TURBO
'ALEM,NORESTE,MOTOR DIESEL,Termica,4482979,2017-07\n',
```

Aclaraciones EJ_02 Los profesores podrán probar la función **archAlst** con distintos valores de `col` y de `ord`, y/o con otros contenidos de `lsArch`.

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

```
def main():
    lsArch = ['central,region,tecnologia,fuente_generacion,generacion_mata_MWh,anio_mes\n',
'CAPE,COMAHUE,TURBO GAS,Termica,218668984,2020-01\n', 'AESP,BUENOS AIRES,TURBO
VAPOR,Termica,193938412,2012-01\n', 'ALEM,NORESTE,MOTOR DIESEL,Termica,4082979,2017-07\n',
'ALICHI,COMAHUE,TURB HIDRAULICA,Hidraulica,100241172,2018-07\n']

    print("\nCASO PRUEBA 01 - col=0,ord=1")
    ls = archAlst (lsArch,col=0,ord=1)
    for sl in ls:
        print(sl)

    print("\nCASO PRUEBA 02 - col=1,ord=0")
    ls = archAlst (lsArch,col=1,ord=0)
    for sl in ls:
        print(sl)
```

CASO PRUEBA 01 - col=0,ord=1

```
['central', 'region', 'tecnologia', 'fuente_generacion', 'generacion_neta_Mwh', 'anio_mes']
['CAPE', 'COMAHUE', 'TURBO GAS', 'Termica', 21868984, '2020-01']
['ALICHI', 'COMAHUE', 'TURB HIDRAULICA', 'Hidraulica', 100241172, '2018-07']
['ALEM', 'NORESTE', 'MOTOR DIESEL', 'Termica', 4082979, '2017-03']
['AESP', 'BUENOS AIRES', 'TURBO VAPOR', 'Termica', 193938412, '2012-01']
```

CASO PRUEBA 02 - col=1,ord=0

```
['central', 'region', 'tecnologia', 'fuente_generacion', 'generacion_neta_Mwh', 'anio_mes']
['AESP', 'BUENOS AIRES', 'TURBO VAPOR', 'Termica', 193938412, '2012-01']
['CAPE', 'COMAHUE', 'TURBO GAS', 'Termica', 21868984, '2020-01']
['ALICHI', 'COMAHUE', 'TURB HIDRAULICA', 'Hidraulica', 100241172, '2018-07']
['ALEM', 'NORESTE', 'MOTOR DIESEL', 'Termica', 4082979, '2017-03']
```

CASO PRUEBA 03 - col=5,ord=1

```
['central', 'region', 'tecnologia', 'fuente_generacion', 'generacion_neta_Mwh', 'anio_mes']
'CAPE', 'COMAHUE', 'TURBO GAS', 'Termica', 21868984, '2020-01']
['ALICHI', 'COMAHUE', 'TURB HIDRAULICA', 'Hidraulica', 100241172, '2018-07']
['ALEM', 'NORESTE', 'MOTOR DIESEL', 'Termica', 4082979, '2017-03']
['AESP', 'BUENOS AIRES', 'TURBO VAPOR', 'Termica', 193938412, '2012-01']
```

CASO PRUEBA 04 - col=100,ord=0

```
['central', 'region', 'tecnologia', 'fuente_generacion', 'generacion_neta_Mwh', 'anio_mes']
['CAPE', 'COMAHUE', 'TURBO GAS', 'Termica', 21868984, '2020-01']
['AESP', 'BUENOS AIRES', 'TURBO VAPOR', 'Termica', 193938412, '2012-01']
['ALEM', 'NORESTE', 'MOTOR DIESEL', 'Termica', 4082979, '2017-03']
['ALICHI', 'COMAHUE', 'TURB HIDRAULICA', 'Hidraulica', 100241172, '2018-07']
```

```
VAPOR,Termica,1735938412,2012-01\n', 'ALEM,NORESTE,MOTOR DIESEL,Termica,4002979,2017-07\n',
'ALICHI,COMAHUE,TURB HIDRAULICA,Hidraulica,1002411172,2018-07\n']

    print("\nCASO PRUEBA 01 - col=0,ord=1")
    ls = archAlst (lsArch,col=0,ord=1)
    for sl in ls:
        print(sl)

    print("\nCASO PRUEBA 02 - col=1,ord=0")
    ls = archAlst (lsArch,col=1,ord=0)
    for sl in ls:
        print(sL)

    print("\nCASO PRUEBA 03 - col=5,ord=1")
    ls = archAlst (lsArch,col=5,ord=1)
    for sl in ls:
        print(sL)

    print("\nCASO PRUEBA 04 - col=100,ord=0")
    ls = archAlst (lsArch,col=100,ord=0)
    for sl in ls:
        print(sL)

main()
```

Al ejecutar el main con pruebas planteado arriba