

APELLIDO Y NOMBRE: _____ **CANTIDAD DE HOJAS:** _____

PARTE I – Complete el programa (26 puntos)

1 - Complete las secciones indicadas con "____" con una línea de código correspondiente. Complete las secciones indicadas con "____(*)____" con una o más líneas de código (todas las que considere necesarias.) El resto de las líneas de código del programa deben quedar sin modificar, y en el orden mostrado:

1.1) - Completar la función **ordenArr** para que ordene de menor a mayor el arreglo **A** de números enteros pasado por parámetro.

```
def ordenArr(A):
    for i in ____:
        for j in ____:
            if(____):
                ____
```

Ej:
Si el arreglo es A es:
[2,7,3,8,1]
Luego de ordenar deberá quedar:
[1, 2, 3, 7, 8]

1.2) - Completar la función **decAbin**, para que retorne la conversión a binario del número decimal **num** recibido por parámetro.

```
def decAbin(num):
    resto=0
    i=____
    bina =0
    while(num>0):
        resto=____
        num=____
        bina=____
        i=____
    return bina
```

Ej:
num= 19, retorna 10011
num=9, retorna 1001

PARTE II – Opción múltiple (26 puntos - Se considera aprobado solamente si están **todos los ejercicios correctos)**

2 - Indique la opción correcta y más completa:

2.1) -

```
def fa(a):
    return fb(a)+1
def fb(b):
    for i in range(0, len(b)):
        return int("".join(b))+i
a=['1','2']
print ("{0:d}{1:d}".format(fb(a),fa(a)))
```

¿Qué imprime al ejecutar el código ?

- a) 1212 b) 1312 c) 1213
d) 1313 e) 1314
f) Ninguna de las anteriores

2.2) -

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

print ("{0:d}{0:d}".format(0,comparar(1,1,1)))
```

¿Qué retorna la función?

- a) 10 b) 11 c) 01 d) 00
e) No imprime nada
f) Ninguna de las anteriores

2.3) -

```
def f1(x):
    for i in range(0,len(x)):
        if(i in x):
            print('a',end="")
        else:
            print('b',end="")
x=list(range(10, 1, 2))
f1(x)
```

¿Qué imprime al ejecutar el código ?

- a) babab b) bbaba c) aaaaa
d) bbbbbb
e) Ninguna de las anteriores

2.4) -

```
s=['Juan','Carlos']
s="Carlos"
s="Juan"
s[0]='x'
print(s)
```

¿Qué imprime al ejecutar el código ?

- a) ['Juan', 'Carlos'] b) ['x', 'Carlos']
c) ['x'] d) [] e) Nada, sale error al ejecutar
f) Ninguna de las anteriores

PARTE III – Programación (48 puntos)

3 – Programación

Una compañía financiera cuenta con información de sus bases de datos, que son los siguientes tres archivos (que no tienen ningún criterio de ordenamiento):

Archivo	Estructura de datos	Ejemplo de contenido
<i>habitantes.txt</i> (archivo CSV)	ID_habitante (entero) nombre (cadena de caracteres) hijos (entero) edad (entero)	1, Juan Cruz, 1, 42 8, María, 3, 55 3, Claudia, 1, 23 9, Diego, 0, 19
<i>localidades.txt</i> (archivo CSV)	ID_localidad (entero) nombre (cadena de caracteres) ID_zona (entero)	1, Adolfo Alsina, 4 6, Carlos Casares, 19 8, Marcos Paz, 23 3, Lujan, 23 7, Ezeiza, 21
<i>habitantesXlocalidad.txt</i> (archivo CSV)	ID_localidad (int) ID_habitante (int)	1, 1 3, 8 7, 9 7, 3 8, 122

Se solicita realizar las siguientes dos funciones:

3.1) - Realizar la función ***cantHabitantes(nombreLocalidad, hijos)*** que reciba como parámetro el nombre de una localidad y la cantidad de hijos, y retorne una lista con los habitantes que hay en dicha localidad que tengan exactamente la cantidad de hijos pasada por parámetro. Cargar en la lista el id del habitante y el nombre.

3.2) - Realizar una función ***edadXlocalidad()*** que imprima en pantalla por cada id de localidad, la edad promedio de los habitantes de dicha localidad, ordenado por id de localidad de menor a mayor.

Ejemplo de salida

Id Localidad	Promedio Edad
1	42,0
3	55,0
7	21,0