



RESPUESTAS A LAS PREGUNTAS TEÓRICAS

EJ.T.1

```
1 def fun(ls,ini):
2     i = ini
3     while i < len(ls):
4         aux=ls.pop(0)
5         ls.append(aux)
6         i+=1
7
8 def main():
9     x=[44,20,14,22,96,12]
10    fun(x,len(x)-1)
11    print(x)
12
13 main()
14
```

¿Qué imprime al ejecutar el código?

- A. [20, 14, 22, 96, 12, 44]
- B. [44, 20, 14, 22, 96, 12]
- C. [14, 22, 96, 12, 44, 20]
- D. [20, 14, 22, 96, 12, None]
- E. [None, None, None, None, None, None]
- F. []
- G. Ninguna de las respuestas son correctas.



EJ.T.2

```
1 def fun (di,nom,parcial,valor):
2     if di.get(nom)!=None:
3         i=di.get(nom)[1]
4         if di[nom][i]["a"].get(parcial)==None:
5             di[nom][i]["a"][parcial]=valor
6
7 def main ():
8     x={"Juan":[{"a":{"pp1":"A","pp2":"D"}},0],
9         "Lola":[{"p1":{"pp2":"A","pp3":"A-"}},1],
10        "Charly":[{"746":{"pp1":"A","pp3":"A-"},},2]
11    }
12    fun(x,"Juan", "pp3", "A")
13    print(x["Juan"])
14
15 main()
```



```
12     fun(x, Juan, pp3, A)
13     print(x["Juan"])
14
15 main()
```

¿Qué imprime al ejecutar el código?

- A. `{'a': {'pp1': 'A', 'pp2': 'D', 'pp3': 'A'}}`
- B. `None`
- C. `[{'a': {'pp1': 'A', 'pp2': 'D', 'pp3': 'A'}}, 0]`
- D. `[{"a":{"pp1":"A","pp2":"D"}},0]`
- E. `{"a":{"pp1":"A","pp2":"D"}}`
- F. Ninguna de las respuestas son correctas.

RESPUESTA CORRECTA:

`[{'a': {'pp1': 'A', 'pp2': 'D', 'pp3': 'A'}}, 0]`

EJ.T.3

```
1 def fun(var):
2     miVar = [50,20,10]
3     aux=-10
4     for i in range(0,len(var)-1):
5         for j in range(i+1,len(var)):
6             if var[i]>var[j]:
7                 var[i] = var[j]
8                 var[j] = var[i]
9
10 def main():
11     miVar = [44,22,11]
12     fun(miVar)
13     print(miVar)
14 main()
```

¿Qué imprime al ejecutar el código?

[-10, -10, -10]

[11, 11, 11]

```
9
10 def main():
11     miVar = [44,22,11]
12     fun(miVar)
13     print(miVar)
14 main()
```

¿Qué imprime al ejecutar el código?

[-10, -10, -10]

[11, 11, 11]

[10, 10, 10]

[50, 20, 10]

[44, 22, 11]

[]

Ninguna de las respuestas son correctas.

RESPUESTA CORRECTA:

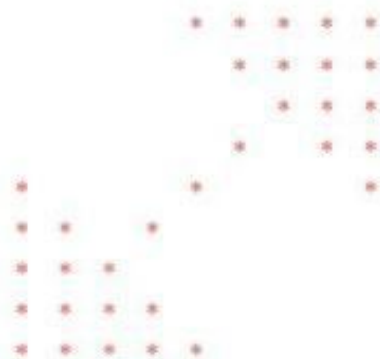
[11, 11, 11]

FUNDAMENTO:



EJ.T.4

```
#####  
##FIGURA A IMPRIMIR ##
```



```
#####
```

```
def figura(b):  
    for f in range(0,b):  
        for c in range(0,b):
```



#####

```
def figura(b):
    for f in range(0,b):
        for c in range(0,b):
            if _____ CONDICIÓN _____ : # <<<<< ¿Cuál es la co
                print(" *", end="")
            else:
                print("  ", end="")
        print()

def main():
    figura(9)
main()
```

¿Qué imprime al ejecutar el código?

- A. $f \leq (c-(b//2))$ or $c \leq (f-(b//2))$ or $f+c==b-1$
- B. $f \leq (c-(b//2))$ and $c \leq (f-(b//2))$ and $f+c==b-1$
- C. $f \leq (c-(b//2))$ or $c \leq (f-(b//2))$ and $f+c==b-1$
- D. $f \leq (c-(b//2))$ and $c \leq (f-(b//2))$ or $f+c==b-1$
- E. $f \leq (c-(b//2))$ or $f+c==b-1$
- F. $f \leq (c-(b//2))$ and $f+c==b-1$
- G. $f \leq (c-(b//2))$ or $c \leq (f-(b//2))$ or $f+c==b-1$ or $f==c$

EJ.T.5

```
1 def fun():
2     lstArch=[]
3     arch = open("arch.csv","r")
4     strLinea = arch.readline()
5     while strLinea!="":
6         strLinea = strLinea[:-1]
7         lstArch.append(strLinea.split(","))
8         strLinea=arch.readline()
9     arch.close()
10    return lstArch
11
12 def main():
13     res=fun()
14     print(res[1][2] + res[2][1])
15 main()
```

¿Qué imprime al ejecutar el código?

33.453

333.45

36.45



¿Qué imprime al ejecutar el código?

33.453

333.45

36.45

33.48

33.33

No imprime, porque hay un ciclo infinito.

No imprime, porque hay algún error de sintaxis en el código.

Imprime, pero no se encuentra el valor dentro de la lista propuesta de posibles valores (en color azul).

RESPUESTA CORRECTA:

33.453

FUNDAMENTO:

La función fun está escrita correctamente. retorna una lista de lista donde cada sublista representa un renglón del archivo.

Todos los elementos de las sublistas son cadena de caracteres (ya que no se realizó ninguna conversión), por lo tanto, Se concatena el 33.45 con el 3:

```
print(res[1][2] + res[2][1])  
print('33.45' + '3')  
print('33.453')
```

