

Comenzado el	domingo, 20 de agosto de 2023, 17:29
Estado	Finalizado
Finalizado en	domingo, 20 de agosto de 2023, 17:58
Tiempo empleado	29 minutos 25 segundos
Puntos	14/14
Calificación	10 de 10 (100%)

Pregunta **1**

Correcta

Se puntúa 1 sobre 1

¿Cuál de las siguientes afirmaciones es CIERTA respecto del uso de índices en un arreglo (representado con una variable de tipo *list*) en Python? Más de una respuesta puede ser válida. Marque todas las que considere correctas.

Seleccione una o más de una:

- ☒ a. Los índices de un arreglo deben ser numéricos, y valen como índices tanto valores negativos, como cero o positivos. ✓
- ☐ b. Se pueden usar índices de tipo *cadena de caracteres* para entrar a una casilla de un arreglo.
- ☒ c. Los índices de un arreglo van numerados de 0 hasta el tamaño del arreglo menos uno. ✓
- ☒ d. En Python, el programador puede hacer que los índices de un arreglo comiencen desde un número mayor a cero. ✗

Pregunta **2**

Correcta

Se puntúa 1 sobre 1

¿Qué hace el siguiente script en Python?

```
n = 20
v = n * [0]
for i in range(n):
    v[i] = 3*i
```

Seleccione una:

- ☐ a. Crea un arreglo unidimensional de 20 componentes, y llena ese arreglo con los números del 0 al 19.
- ☒ b. Crea un arreglo unidimensional de 20 componentes, y llena ese arreglo con los primeros 20 múltiplos de 3. ✓
- ☐ c. Crea un arreglo unidimensional de 20 componentes, y llena ese arreglo con 20 números aleatorios.
- ☐ d. Crea un arreglo unidimensional de 20 componentes, y llena ese arreglo con el número 3 (veinte veces el 3).

Pregunta 3

Correcta

Se puntúa 1 sobre 1

¿Cuál de las siguientes afirmaciones es CIERTA respecto del uso de arreglos (representados con variables de tipo *list*) en Python? Más de una respuesta puede ser válida. Marque todas las que considere correctas.

Seleccione una o más de una:

- ☒ a. Los arreglos implementados como variables de tipo *list* en Python pueden aumentar o disminuir su tamaño a medida que el programador lo requiera. ✓
- ☐ b. Un arreglo implementado como una variable de tipo *list* es completamente equivalente a una *tuple* en Python.
- ☒ c. Siempre se puede saber el tamaño de un arreglo llamando a la función *len()* provista por Python. ✓
- ☒ d. El primer índice de cada dimensión de un arreglo en Python es siempre cero (a menos que se usen índices negativos). ✓

Pregunta 4

Correcta

Se puntúa 1 sobre 1

Analice el siguiente script en Python:

```
n = 6
v = n * [0]
for i in range(n):
    v[i] = '123'
```

¿Es correcto este script, o existe algún problema con él?

Seleccione una:

- ☒ a. No hay nada de malo con ese segmento. ✓
- ☐ b. El arreglo fue creado como un arreglo de *n* valores de tipo *int*, y por lo tanto no pueden asignarse luego cadenas de caracteres en sus casilleros.
- ☐ c. El arreglo *v* está mal definido: deben usarse paréntesis (o sea: $v = n * (0)$) en lugar de corchetes (o sea, en lugar de: $v = n * [0]$)
- ☐ d. Lanza un error de índice fuera de rango y se interrumpe, pues se intenta acceder a una casilla con índice fuera de rango en el ciclo *for*.

Pregunta 5

Correcta

Se puntúa 1 sobre 1

Analice el siguiente script en Python:

```
n = 6
v = n * [0]
for i in range(n+1):
    v[i] = i
```

¿Hay algún problema con el script mostrado?

Seleccione una:

- ☐ a. El arreglo está mal definido: la instrucción $v = n * [0]$ está creando un arreglo vacío, sin casilleros.
- ☒ b. Lanza un error y se interrumpe, pues en el ciclo *for* se intenta acceder a una casilla no definida. ✓
- ☐ c. Convierte cada casilla del vector a una cadena de caracteres.
- ☐ d. No hay nada de malo con ese script.

Pregunta **6**

Correcta

Se puntúa 1 sobre 1

Analice el siguiente script en Python:

```
v = [2, 4, 1, 6]
v[0] = v[v[0]] * 3
print('v[0]:', v[0])
```

¿Cuál de las siguientes es correcta en relación al script mostrado?

Seleccione una:

- ☐ a. Se mostrará el mensaje: $v[0] = 2$.
- ☒ b. Se mostrará el mensaje: $v[0] = 3$. ✓
- ☐ c. Se lanzará un error y el script se interrumpirá.
- ☐ d. Se mostrará el mensaje: $v[0] = 12$.

Pregunta **7**

Correcta

Se puntúa 1 sobre 1

Analice el siguiente script en Python:

```
n = 5
a = n * [0]
for i in range(n):
    a[i] = i + 1

v = n * [0]
for i in range(n):
    v[a[i]] = a[i]
```

¿Cuál de las siguientes es correcta en relación al script mostrado?

Seleccione una:

- ☐ a. Todas las casillas del arreglo a se convierten a *float* y se vuelven a asignar en el mismo arreglo a .
- ☐ b. El arreglo v queda valiendo los mismos valores que el arreglo a , pero convertidos al tipo *float*.
- ☒ c. El script lanza un error y se interrumpe por intentar acceder a una casilla fuera de rango en el arreglo v . ✓
- ☐ d. Tanto el arreglo a como el v quedan con todos sus casilleros valiendo 0.

Pregunta 8

Correcta

Se puntúa 2 sobre 2

¿Qué hace el siguiente programa en Python?

```
__author__ = 'Cátedra de AED'

def test():
    n = 10
    v = n * [0]

    for i in range(n):
        v[i] = int(input('v[' + str(i) + ']: '))

    im = 0
    for i in range(1, n):
        if v[i] < v[im]:
            im = i

    print('El valor pedido es:', v[im])

if __name__ == '__main__':
    test()
```

Seleccione una:

- ☐ a. Carga por teclado un arreglo unidimensional con 10 números enteros. Luego calcula y muestra el promedio de los elementos contenidos en ese arreglo.
- ☒ b. Carga por teclado un arreglo unidimensional con 10 números enteros. Luego busca el menor valor contenido en el arreglo y muestra ese menor. ✓
- ☐ c. Carga por teclado un arreglo unidimensional con 10 números enteros. Luego busca el mayor valor contenido en el arreglo y muestra ese mayor.
- ☐ d. Carga por teclado un arreglo con 10 números enteros y luego ordena y muestra ese arreglo.

Pregunta 9

Correcta

Se puntúa 2 sobre 2

¿Qué hace la siguiente función en Python?

```
def comprobar(v):
    n = len(v)
    for i in range(n-1):
        if v[i] > v[i+1]:
            return False
    return True
```

Seleccione una:

- ☒ a. Retorna **True** si el arreglo **v** tomado como parámetro está ordenado de menor a mayor, o retorna **False** en caso contrario. ✓
- ☐ b. Retorna **True** si el arreglo **v** tomado como parámetro está ordenado de mayor a menor, o retorna **False** en caso contrario.
- ☐ c. Retorna **True** si el arreglo **v** tomado como parámetro contiene al valor **n**, o retorna **False** en caso contrario.
- ☐ d. Retorna **True** si el arreglo **v** tomado como parámetro contiene todos sus elementos iguales, o retorna **False** en caso contrario.

¿Qué hace la siguiente función en Python?

```
def generar(v):  
    n = len(v)  
  
    ac = 0  
    for i in range(n):  
        ac += v[i]  
    p = ac / n  
  
    c = 0  
    for i in range(n):  
        if v[i] >= p:  
            c += 1  
  
    mp = c * [0]  
    idx = 0  
    for i in range(n):  
        if v[i] >= p:  
            mp[idx] = v[i]  
            idx += 1  
  
    return mp
```

Seleccione una:

- ☐ a. Toma un arreglo **v** como parámetro. Genera un segundo arreglo **mp** que contiene sólo los elementos de **v** que son mayores a 10 y retorna el nuevo arreglo **mp**.
- ☒ b. Toma un arreglo **v** como parámetro. Calcula el promedio de los valores contenidos en **v**. Finalmente, genera un segundo arreglo **mp** que contiene sólo los elementos de **v** que son mayores o iguales al promedio y retorna el nuevo arreglo **mp**. ✓
- ☐ c. Toma un arreglo **v** como parámetro. Ordena el arreglo **v**. Finalmente, genera un segundo arreglo **mp** que contiene todos los elementos de **v**, y retorna **mp** (que entonces, será igual al vector **v** pero ordenado).
- ☐ d. Toma un arreglo **v** como parámetro. Genera un segundo arreglo **mp** que contiene solo los elementos no negativos de **v**, y retorna el nuevo arreglo **mp**.

◀ Materiales Adicionales para la Ficha 15

Ir a...



Guía 15 de Ejercicios Prácticos ▶