Comenzado el	domingo, 29 de julio de 2018, 11:01
Estado	Finalizado
Finalizado en	lunes, 30 de julio de 2018, 06:59
Tiempo empleado	19 horas 58 minutos
Puntos	14/14
Calificación	**10** de 10 (**100**%)

Pregunta 1

Correcta

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:

- [x] a.
  Los índices de un arreglo van numerados de 0 hasta el tamaño del arreglo menos uno. ✓

  ¡Correcto!
- [x] b.
  Los índices de un arreglo deben ser numéricos, y valen como índices tanto valores negativos, como cero o positivos. ✓

  ¡Correcto!
- [ ] c.
  En Python, el programador puede hacer que los índices de un arreglo comiencen desde un número mayor a cero.
- [ ] d.
  Se pueden usar índices de tipo *cadena de caracteres* para entrar a una casilla de un arreglo.

¡Correcto!

Las respuestas correctas son:
Los índices de un arreglo deben ser numéricos, y valen como índices tanto valores negativos, como cero o positivos.,

Los índices de un arreglo van numerados de 0 hasta el tamaño del arreglo menos uno.

Pregunta 2

Correcta

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. ✓

¡Correcto!

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).

¡Correcto!

La respuesta correcta es:
Crea un arreglo unidimensional de 20 componentes, y llena ese arreglo con los primeros 20 múltiplos de 3.

Pregunta 3

Correcta

Puntúa 1 sobre 1

¿Cuál de las siguientes afirmaciones es CIERTA respecto del uso 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:

- [x] a.
  El primer índice de cada dimensión de un arreglo en Python es siempre cero (a menos que se usen índices negativos). ✓

  ¡Correcto!

- [x] b.
  Los arreglos implementados como variables de tipo *list* en Python pueden aumentar o disminuir su tamaño a medida que el programador lo requiera. ✓

  ¡Correcto!

- [ ] c.
  Un arreglo implementado como una variable de tipo *list* es completamente equivalente a una *tuple* en Python.

- [x] d.
  Siempre se puede saber el tamaño de un arreglo llamando a la función *len()* provista por Python. ✓

  ¡Correcto!

¡Correcto!

Las respuestas correctas son:
Los arreglos implementados como variables de tipo *list* en Python pueden aumentar o disminuir su tamaño a medida que el programador lo requiera.,

Siempre se puede saber el tamaño de un arreglo llamando a la función *len()* provista por Python.,

El primer índice de cada dimensión de un arreglo en Python es siempre cero (a menos que se usen índices negativos).

Pregunta 4

Correcta

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.
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.

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.
No hay nada de malo con ese segmento. ✓

¡Correcto! Efectivamente, el arreglo comienza con seis ceros, y luego se cambian esos seis ceros por la cadena '123' repetida seis veces... Y esto es válido por ser Python un lenguaje de tipado dinámico.

d.
El arreglo *v* está mal definido: deben usarse parántesis (o sea: *v = n * (0)*) en lugar de corchetes (o sea, en lugar de: *v = n * [0]*)

¡Correcto!

La respuesta correcta es:
No hay nada de malo con ese segmento.

Pregunta 5

Correcta

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.
No hay nada de malo con ese script.

b.
El arreglo está mal definido: la instrucción *v = n * [0]* está creando un arreglo vacío, sin casilleros.

c.
Lanza un error y se interrumpe, pues en el ciclo *for* se intenta acceder a una casilla no definida. ✓

¡Correcto! En efecto, el ciclo for está incluyendo una vuelta con i = 6... y el índice de la última casilla es 5...

d.
Convierte cada casilla del vector a una cadena de caracteres.

¡Correcto!

La respuesta correcta es:
Lanza un error y se interrumpe, pues en el ciclo *for* se intenta acceder a una casilla no definida.

Pregunta 6

Correcta

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 siguentes es correcta en relación al script mostrado?

Seleccione una:

a.
Se mostrará el mensaje: *v[0] = 12*.

b.
Se mostrará el mensaje: *v[0] = 2*.

c.
Se lanzará un error y el script se interrumpirá.

d.
Se mostrará el mensaje: *v[0] = 3*.

¡Correcto!

¡Correcto!

La respuesta correcta es:
Se mostrará el mensaje: *v[0] = 3*.

Pregunta 7

Correcta

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 siguentes es correcta en relación al script mostrado?

Seleccione una:

a.
El arreglo *v* queda valiendo los mismos valores que el arreglo *a*, pero convertidos al tipo *float*.

b.
Tanto el arreglo *a* como el *v* quedan con todos sus casilleros valiendo 0.

c.
El script lanza un error y se interrumpe por intentar acceder a una casilla fuera de rango en el arreglo *v*. ✓

¡Correcto! Los valores de los casilleros de *a* se están usando como índices para entrar en *v*... pero el último casillero de *a* está valiendo 5, con lo cual se produce un error en la última vuelta del segundo *for*.

d.
Todos las casillas del arreglo *a* se convierten a *float* y se vuelven a asignar en el mismo arreglo *a*.

¡Correcto!

La respuesta correcta es:
El script lanza un error y se interrumpe por intentar acceder a una casilla fuera de rango en el arreglo *v*.

Pregunta 8

Correcta

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 busca el mayor valor contenido en el arreglo y muestra ese mayor.

b.
Carga por teclado un arreglo unidimensional con 10 números enteros. Luego calcula y muestra el promedio de los elementos contenidos en ese arreglo.

c.
Carga por teclado un arreglo con 10 números enteros y luego ordena y muestra ese arreglo.

d.
Carga por teclado un arreglo unidimensional con 10 números enteros. Luego busca el menor valor contenido en el arreglo y muestra ese menor. ✓

¡Correcto!

¡Correcto!

La respuesta correcta es:

Carga por teclado un arreglo unidimensional con 10 números enteros. Luego busca el menor valor contenido en el arreglo y muestra ese menor.

Pregunta 9

Correcta

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 contiene al valor *n*, 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 todos sus elementos iguales, o retorna *False* en caso contrario.

d.
Retorna *True* si el arreglo *v* tomado como parámetro está ordenado de menor a mayor, o retorna *False* en caso contrario. ✓

¡Correcto!

¡Correcto!

La respuesta correcta es:
Retorna *True* si el arreglo *v* tomado como parámetro está ordenado de menor a mayor, o retorna *False* en caso contrario.

Pregunta 10

Correcta

Puntúa 3 sobre 3

¿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. 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).

c.
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.

d.
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. ✓

¡Correcto!

¡Correcto!

La respuesta correcta es:
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.