

Comenzado el lunes, 10 de septiembre de 2018, 14:18

Estado Finalizado

Finalizado en lunes, 10 de septiembre de 2018, 14:41

Tiempo empleado 22 minutos 45 segundos

Puntos 13/15

Calificación 9 de 10 (87%)

Pregunta 1

Correcta

Puntúa 1 sobre 1

¿Cuál de la siguientes afirmaciones es *incorrecta* en relación a las características de los arreglos en Python?

Seleccione una:

- ☐ a.
Un arreglo (variable de tipo *list*) en Python puede contener valores de cualquier tipo, incluso entremezclados.
- ☐ b.
Siempre se puede usar la función `len()` para determinar la longitud o cantidad de componentes de un arreglo (variable de tipo *list*)
- ☒ c.
En Python sólo se pueden crear arreglos (variables de tipo *list*) de dimensión 1 o de dimensión 2. ✓
¡Correcto! Esta afirmación es justamente falsa: en Python una variable de tipo *list* puede contener a otras anidadas, permitiendo que el programador cree así arreglos de la dimensión que necesite.
- ☐ d.
Los índices de un arreglo (variable de tipo *list*) en Python, comienzan siempre desde el número 0 (si no se consideran índices negativos)

¡Correcto!

La respuesta correcta es:

En Python sólo se pueden crear arreglos (variables de tipo *list*) de dimensión 1 o de dimensión 2.

Pregunta 2

Correcta

Puntúa 2 sobre 2

Asuma que se ha creado una matriz *reg* en la que cada fila representa un año calendario desde el 2000 en adelante (la fila 0 representa al año 2000, la 1 al 2001 y así sucesivamente) y cada columna representa una de las carreras disponibles (numeradas desde 0 en adelante). Suponga que en casillero *reg[i][j]* se ha almacenado un número que indica cuántos alumnos se inscribieron en la materia *j* en el año *i*, y suponga finalmente que esa matriz fue correctamente creada y cargada. ¿Qué hace la siguiente función, suponiendo que toma como parámetro a la citada matriz *reg*?

```
def procesar(reg):
    anios = len(reg)
    carreras = len(reg[0])
    fm, cm = 0, 0
    for i in range(anios):
        for j in range(carreras):
            if reg[i][j] > rge[fm][cm]:
                fm = i
                cm = j
    return fm
```

Seleccione una:

- ☐ a.
Busca en toda la matriz el casillero con el mayor número de inscriptos y retorna el índice de columna que corresponde a ese casillero (o sea, retorna el número de la carrera para la que ocurrió ese caso).
- ☐ b.
Busca en toda la matriz el casillero con el menor número de inscriptos y retorna el índice de fila que corresponde a ese casillero (o sea, retorna el año en que ocurrió ese caso).
- ☐ c.
Busca en toda la matriz el casillero que tenga el menor número de inscriptos y retorna el índice de columna que corresponde a ese casillero (o sea, retorna el número de la carrera para la que ocurrió ese caso).
- ☒ d.
Busca en toda la matriz el casillero con el mayor número de inscriptos y retorna el índice de fila que corresponde a ese casillero (o sea, retorna el año en que ocurrió ese caso). ✓

¡Correcto!

¡Correcto!

La respuesta correcta es:

Busca en toda la matriz el casillero con el mayor número de inscriptos y retorna el índice de fila que corresponde a ese casillero (o sea, retorna el año en que ocurrió ese caso).

Pregunta 3

Correcta

Puntúa 1 sobre 1

¿Qué hace el siguiente segmento de instrucciones en Python?

```
p = []
for i in range(3):
    p.append([])
    for j in range(2):
        p[i].append([])
        for k in range(4):
            p[i][j].append([])

p[2][1][3] = 'Prueba'
print(p)
```

Seleccione una:

- ☒ a.
Crea un arreglo *p* de tres dimensiones, y almacena una cadena en el último casillero de ese arreglo (el casillero de índices más altos de cada dimensión posible). ✓
¡Correcto!
- ☐ b.
Crea un arreglo *p* de cuatro dimensiones, y almacena una cadena en el último casillero de ese arreglo (el casillero de índices más altos de cada dimensión posible).
- ☐ c.
Crea un arreglo *p* de tres dimensiones, y almacena una cadena en el primer casillero de ese arreglo (el casillero de índices más bajos de cada dimensión posible).
- ☐ d.
Lanza un error de intérprete: en Python no se pueden crear arreglos (variables de tipo *list*) de dimensión mayor a 2.

¡Correcto!

La respuesta correcta es:

Crea un arreglo *p* de tres dimensiones, y almacena una cadena en el último casillero de ese arreglo (el casillero de índices más altos de cada dimensión posible).

Pregunta 4

Incorrecta

Puntúa 0 sobre 1

Suponga la función `test()` (que se muestra más abajo) recibe como parámetro un arreglo bidimensional *mat* ya creado con componentes de tipo *int*.

¿Qué hace concretamente esa función `test()`?

```
def test(mat):  
    fils = len(mat)  
    cols = len(mat[0])  
    for k in range(cols):  
        ac = 0  
        for g in range(fils):  
            ac += mat[k][g]  
        p = ac / fils  
        print('Indice:', k, '- Promedio:', p)
```

Seleccione una:

☐ a.

Lo que ocurre con este método depende de la cantidad *n* de filas y y la cantidad *m* de columnas que tiene la matriz. Si la matriz es cuadrada ($n == m$) entonces mostrará por pantalla un listado con el promedio de cada *columna*. Si la matriz no es cuadrada ($n != m$) entonces al ejecutar este método se *provocará una excepción por índice fuera de rango*.

☐ b.

Lo que ocurre con este método depende de la cantidad *n* de filas y y la cantidad *m* de columnas que tiene la matriz. Si la matriz es cuadrada ($n == m$) entonces mostrará por pantalla un listado con el promedio de cada *fila*. Si la matriz no es cuadrada ($n != m$) entonces al ejecutar este método se *provocará una excepción por índice fuera de rango*.

☐ c.

Muestra por pantalla un listado con el promedio de cada fila.

☒ d.

Muestra por pantalla un listado con el promedio de cada columna. ✖

Incorrecto... puede llegar a a hacer eso... pero depende de un análisis un poco más profundo...

Incorrecto... revise las fichas de clase sobre arreglos de una o más dimensiones y sus notas de clase... Revise lo que ocurre cuando intenta acceder a un casillero que no existe...

La respuesta correcta es:

Lo que ocurre con este método depende de la cantidad *n* de filas y y la cantidad *m* de columnas que tiene la matriz. Si la matriz es cuadrada ($n == m$) entonces mostrará por pantalla un listado con el promedio de cada *fila*. Si la matriz no es cuadrada ($n != m$) entonces al ejecutar este método se *provocará una excepción por índice fuera de rango*.

Pregunta 5


Correcta

Puntúa 2 sobre 2

Asuma que se quiere definir una función `registrar()` como la que se muestra más abajo, en la cual se necesita crear una matriz `reg` de forma que en esa matriz cada fila represente un año calendario desde el 2000 en adelante (la fila 0 representaría al año 2000, la 1 al 2001 y así sucesivamente) y cada columna represente una de las carreras disponibles en una universidad (numeradas desde 0 en adelante). La idea es que una vez creada esa matriz, debe usarse para almacenar en cada casillero una cierta cantidad fija de inscriptos ¿Hay algún problema con la siguiente función en Python, asumiendo que **a** es la cantidad de filas y **c** es la cantidad de columnas?

```
def registrar(a, c):  
    reg = []  
    for i in range(a):  
        for j in range(c):  
            reg[i][j] = 50  
  
    return reg
```

Seleccione una:

- ☐ a.
No, no hay problema alguno. La función crea la matriz pedida, e inicializa cada casillero con el valor 50.
- ☐ b.
Sí, hay un problema: los índices *i* y *j* están al revés cuando se usaron como índices para entrar a una casilla de la matriz.
- ☒ c.
Sí, hay un problema: al ejecutar la función de producirá un error de intérprete y el programa se interrumpirá, debido a que la matriz en realidad nunca fue creada.

¡Correcto!
- ☐ d.
Sí, hay un problema pero es menor: la función crea efectivamente la matriz pedida, pero luego inicializa cada casillero con el mismo valor, lo cual no tiene sentido práctico.

¡Correcto!

La respuesta correcta es:

Sí, hay un problema: al ejecutar la función de producirá un error de intérprete y el programa se interrumpirá, debido a que la matriz en realidad nunca fue creada.

Pregunta 6

Correcta

Puntúa 2 sobre 2

Suponga *función test()* (que se muestra más abajo) toma como parámetro una matriz *mat* ya definida (y otras dos variables *x* e *y*) y correctamente cargada, con elementos de tipo *int*. ¿Qué hace concretamente esta función?

```
def test(mat, x, y):  
    n = len(mat)  
    m = len(mat[0])  
    if x < 0 or x >= n:  
        return False  
    if y < 0 or y >= n:  
        return False  
    if x == y:  
        return True  
  
    for k in range(m):  
        mat[x][k], mat[y][k] = mat[y][k], mat[x][k]  
  
    return True
```

Seleccione una:

- ☐ a.
Intercambia los valores contenidos en la columna *x* con los valores contenidos en la columna *y*. Retorna *True* si el intercambio pudo hacerse (o si *x* era igual a *y*); y retorna *False* en caso contrario.
- ☐ b.
Comprueba si alguno de los valores contenidos en la fila *x* es igual a alguno de los valores contenidos en la fila *y*. Retorna *True* en ese caso (o si *x* era igual a *y*); y retorna *False* en caso contrario.
- ☒ c.
Intercambia los valores contenidos en la fila *x* con los valores contenidos en la fila *y*. Retorna *True* si el intercambio pudo hacerse (o si *x* era igual a *y*); y retorna *False* en caso contrario. ✓
¡Correcto!
- ☐ d.
Comprueba si la matriz es cuadrada. En caso de serlo, intercambia los valores contenidos en la fila *x* con los valores contenidos en la columna *y*. Retorna *True* si el intercambio pudo hacerse y retorna *False* en caso contrario.

¡Correcto!

La respuesta correcta es:

Intercambia los valores contenidos en la fila *x* con los valores contenidos en la fila *y*. Retorna *True* si el intercambio pudo hacerse (o si *x* era igual a *y*); y retorna *False* en caso contrario.

Pregunta 7

Correcta

Puntúa 2 sobre 2

Suponga la función *test()* (que se muestra más abajo) que toma como parámetro una matriz *mat* ya definida y cargada con elementos de tipo *int*. ¿Qué hace concretamente la función *test()*?

```
def test(mat):  
    n = len(mat)  
    m = len(mat[0])  
    r = [n*[0] for k in range(m)]  
  
    for i in range(n):  
        for j in range(m):  
            r[j][i] = mat[i][j]  
  
    return r
```

Seleccione una:

- ☐ a.
Retorna una copia de la matriz *mat*.
- ☐ b.
Retorna la *máxima matriz cuadrada que puede obtenerse a partir de mat*. Si *mat* ya era cuadrada, retorna una copia de la misma matriz *mat*. Si *mat* era rectangular, crea una nueva matriz con la máxima cantidad de filas y columnas de *mat* que puedan usarse para que esta nueva matriz sea cuadrada, y retorna esa matriz.
- ☒ c.
Obtiene y retorna la *matriz traspuesta* de *mat* (la *traspuesta* de una matriz *mat* de orden $n*m$ es igual a otra matriz *r* de orden $m*n$, pero de forma tal que las filas de *mat* son ahora las columnas de *r*). ✓
¡Correcto!
- ☐ d.
Retorna *una nueva matriz, en la cual los valores originales de las filas de mat se invierten*: en cada fila, el valor que en *mat* estaba en la columna 0 aparece en la última columna de la nueva matriz, el valor que estaba en la columna 1 de *mat* aparece en la anteúltima de la nueva matriz, y así sucesivamente.

¡Correcto!

La respuesta correcta es:

Obtiene y retorna la *matriz traspuesta* de *mat* (la *traspuesta* de una matriz *mat* de orden $n*m$ es igual a otra matriz *r* de orden $m*n$, pero de forma tal que las filas de *mat* son ahora las columnas de *r*).

Pregunta 8

Correcta

Puntúa 2 sobre 2

Suponga la función *test()* (que se muestra más abajo) que toma como parámetro una matriz *mat* ya creada y cargada con elementos *int*. ¿Qué hace concretamente *la función test()*?

```
def test(mat):  
    n = len(mat)  
    m = len(mat[0])  
  
    r = m * [0]  
    for i in range(m):  
        ac = 0  
        for j in range(n):  
            ac += mat[j][i]  
        r[i] = ac / n  
  
    return r
```

Seleccione una:

- ☐ a.
Obtiene el valor promedio de cada fila, y retorna un vector con esos promedios.
- ☒ b.
Obtiene el valor promedio de cada columna, y retorna un vector con esos promedios. ✓
¡Correcto!
- ☐ c.
Obtiene el valor mayor de cada columna, y retorna un vector con esos mayores.
- ☐ d.
Obtiene el valor promedio de la matriz, y retorna ese valor.

¡Correcto!

La respuesta correcta es:

Obtiene el valor promedio de cada columna, y retorna un vector con esos promedios.

Pregunta 9

Incorrecta

Puntúa 0 sobre 1

Suponga la función *test()* (que se muestra más abajo) que toma como parámetro una matriz *mat* ya creada con valores de tipo *int*. ¿Qué hace concretamente función?

```
def test(mat):  
    n = len(mat)  
    m = len(mat[0])  
  
    if n != m:  
        return False  
  
    for i in range(n):  
        if mat[i][i] != 0:  
            return False  
  
    return True
```

Seleccione una:

- ☐ a.
Comprueba si la matriz es cuadrada. En caso de serlo, retorna *True* si todos los casilleros de esa matriz valen *0(cero)*. Retorna *False* si la matriz no es cuadrada o si alguno de los casilleros de la matriz contiene un valor diferente a cero.
- ☐ b.
Comprueba si la matriz es cuadrada. Retorna *True* si lo es, o retorna *False* en caso contrario.
- ☒ c.
Comprueba si la matriz es cuadrada. En caso de serlo, retorna *True* si la diagonal principal de esa matriz contiene todos sus componentes valiendo distinto de *0(cero)*. Retorna *False* si la matriz no es cuadrada o si alguno de los casilleros de la diagonal contiene el valor cero. ❌

Incorrecto... es cierto que comprueba algo en la diagonal, pero no exactamente lo que marqué aquí...
- ☐ d.
Comprueba si la matriz es cuadrada. En caso de serlo, retorna *True* si la diagonal principal de esa matriz contiene todos sus componentes valiendo *0(cero)*. Retorna *False* si la matriz no es cuadrada o si alguno de los casilleros de la diagonal contiene un valor diferente a cero.

Revise la Fichas 16, sección de Temas Avanzados...

La respuesta correcta es:

Comprueba si la matriz es cuadrada. En caso de serlo, retorna *True* si la diagonal principal de esa matriz contiene todos sus componentes valiendo *0(cero)*. Retorna *False* si la matriz no es cuadrada o si alguno de los casilleros de la diagonal contiene un valor diferente a cero.

Pregunta 10

Correcta

Puntúa 1 sobre 1

Suponga la función `test()` (que se muestra más abajo) que recibe como parámetro una matriz `mat` de elementos de tipo `int` (además de otro parámetro `x`) ¿Qué hace concretamente la función?

```
def test(mat, x):  
    n = len(mat)  
    m = len(mat[0])  
  
    if x < 0 or x >= n:  
        print('Imposible mostrar lo pedido...')  
        return  
  
    for k in range(m):  
        print('Valor:', mat[x][k])
```

Seleccione una:

- ☒ a.
Muestra en consola de salida la fila cuyo índice es `x`. ✓
¡Correcto!
- ☐ b.
Muestra en consola de salida la columna cuyo índice es `x`.
- ☐ c.
Muestra en consola de salida la matriz completa.
- ☐ d.
Muestra en consola de salida la última fila de la matriz.

¡Correcto!

La respuesta correcta es:

Muestra en consola de salida la fila cuyo índice es `x`.