

Comenzado el	domingo, 3 de septiembre de 2023, 20:37
Estado	Finalizado
Finalizado en	domingo, 3 de septiembre de 2023, 20:47
Tiempo empleado	9 minutos 53 segundos
Puntos	14/14
Calificación	10 de 10 (100%)

Pregunta **1**

Correcta

Se puntúa 1 sobre 1

El siguiente programa crea tres arreglos paralelos para almacenar los legajos, los nombres y los promedios de n estudiantes y luego los carga por teclado. ¿Qué muestra finalmente este programa al ejecutarlo?

```
_author__ = 'Cátedra de AED'

def read(leg, nom, pro):
    n = len(leg)
    for i in range(n):
        leg[i] = int(input('Legajo: '))
        nom[i] = input('Nombre: ')
        pro[i] = float(input('Promedio: '))

def procesar(leg, nom, pro):
    im = 0
    for i in range(1, len(pro)):
        if pro[i] > pro[im]:
            im = i
    return leg[im], nom[im], pro[im]

def test():
    n = int(input('Cantidad de alumnos: '))
    leg = n * [0]
    nom = n * ['']
    pro = n * [0.0]
    read(leg, nom, pro)

    r = procesar(leg, nom, pro)

    print('Datos del estudiante pedido...')
    print('Legajo:', r[0])
    print('Nombre:', r[1])
    print('Promedio:', r[2])

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

Seleccione una:

- ☐ a. Busca el estudiante con mayor promedio del arreglo (mediante la función *procesar()*) y muestra el índice de la casilla del arreglo que lo contiene.
- ☒ b. Busca el estudiante con mayor promedio del arreglo (mediante la función *procesar()*) y muestra sus datos completos. ✓
- ☐ c. Busca el estudiante con menor promedio del arreglo (mediante la función *procesar()*) y muestra sus datos completos.
- ☐ d. Busca el estudiante con mayor promedio del arreglo (mediante la función *procesar()*) y muestra sólo el valor de ese promedio.

Pregunta **2**

Correcta

Se puntúa 1 sobre 1

El siguiente programa crea y carga un arreglo *numeros* con *n* números enteros y luego procesa ese arreglo con la función `control()` ¿Qué hace exactamente esa función al ejecutarse?

```
__author__ = 'Cátedra de AED'

def read(numeros):
    n = len(numeros)
    for i in range(n):
        numeros[i] = int(input('Valor[' + str(i) + ']: '))

def control(numeros):
    n = len(numeros)
    for i in range(n-1):
        if numeros[i] != numeros[i+1]:
            return False
    return True

def test():
    n = int(input('Cantidad de números a cargar: '))
    numeros = n * [0]
    read(numeros)

    print()
    if control(numeros):
        print('El contenido del arreglo es correcto')
    else:
        print('El contenido del arreglo no es correcto')

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

Seleccione una:

- ☐ a. Retorna True si los números del arreglo no son todos iguales (hay dos o más números diferentes).
- ☐ b. Retorna True si el arreglo está ordenado de menor a mayor.
- ☐ c. Retorna True si el arreglo está ordenado de mayor a menor.
- ☒ d. Retorna True si todos los números del arreglo son iguales. ✓

Pregunta 3

Correcta

Se puntúa 1 sobre 1

El siguiente programa crea y carga un arreglo con n números enteros y luego procesa ese arreglo mediante las funciones `conteo()` y `suma()`. ¿Qué hacen exactamente esas dos funciones?

```
__author__ = 'Cátedra de AED'

def read(numeros):
    n = len(numeros)
    for i in range(n):
        numeros[i] = int(input('Valor para la casilla ' + str(i) + ': '))

def conteo(numeros):
    c = 0
    for x in numeros:
        if x > 0:
            c += 1
    return c

def suma(numeros):
    s = 0
    n = len(numeros)
    for i in range(0, n, 2):
        s += numeros[i]
    return s

def test():
    n = int(input('Cantidad de números a cargar: '))
    numeros = n * [0]
    read(numeros)

    cmc = conteo(numeros)
    scp = suma(numeros)

    print('Conteo pedido:', cmc)
    print('Suma pedida:', scp)

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

Seleccione una:

- ☐ a. La función `conteo()` cuenta los números mayores a 0 almacenados en casillas con índice impar que hay en el arreglo `numeros`, y la función `suma()` acumula los valores contenidos en las casillas con índice par del arreglo `numeros`.
- ☒ b. La función `conteo()` cuenta los números mayores a 0 que hay en el arreglo `numeros`, y la función `suma()` acumula los valores contenidos en las casillas con índice par del arreglo `numeros`. ✓
- ☐ c. La función `conteo()` cuenta los números mayores a 0 que hay en el arreglo `numeros`, y la función `suma()` acumula los valores contenidos en las casillas con índice impar del arreglo `numeros`.
- ☐ d. La función `conteo()` cuenta los números mayores a 0 que hay en el arreglo `numeros`, y la función `suma()` acumula los valores del arreglo `numeros`.

El siguiente programa crea y carga un arreglo *ventas* con *n* montos de ventas realizadas por un comercio en distintos momentos. ¿Qué hace exactamente el programa completo al ejecutarse?

```
__author__ = 'Cátedra de AED'

def read(ventas):
    n = len(ventas)
    for i in range(n):
        ventas[i] = int(input('Monto de venta[' + str(i) + ']: '))

def sort(ventas):
    n = len(ventas)
    for i in range(n-1):
        for j in range(i+1, n):
            if ventas[i] < ventas[j]:
                ventas[i], ventas[j] = ventas[j], ventas[i]

def display(ventas, cant):
    n = len(ventas)
    if cant > n:
        print('La cantidad de ventas registradas no alcanza para el listado pedido...')
        return

    print('Montos de las', cant, 'ventas pedidas:')
    for i in range(cant):
        print('Monto[' + str(i) + ']:', ventas[i])

def test():
    n = int(input('Cantidad de ventas a cargar: '))
    ventas = n * [0.0]
    read(ventas)

    sort(ventas)

    print()
    display(ventas, 3)

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

Seleccione una:

- ☐ a. Muestra los *cant* mayores montos del arreglo *ventas*, ordenados de menor a mayor.
- ☒ b. Muestra los *cant* mayores montos del arreglo *ventas*, ordenados de mayor a menor. ✓
- ☐ c. Muestra siempre todos los montos del arreglo *ventas*, ordenados de mayor a menor.
- ☐ d. Muestra siempre todos los montos del arreglo *ventas*, en el mismo orden que tenían en el arreglo *ventas*.

Pregunta 5

Correcta

Se puntúa 2 sobre 2

El siguiente programa crea y carga un arreglo *temp* con *n* valores de temperaturas medidas en diferentes momentos y luego procesa ese arreglo mediante la función *amplitud()* ¿Qué hace exactamente esa función?

```
__author__ = 'Cátedra de AED'

def read(temp):
    n = len(temp)
    for i in range(n):
        temp[i] = int(input('Temperatura[' + str(i) + ']: '))

def amplitud(temp):
    n = len(temp)
    my = mn = temp[0]
    for i in range(1, n):
        if temp[i] > my:
            my = temp[i]
        elif temp[i] < mn:
            mn = temp[i]

    return my - mn

def test():
    n = int(input('Cantidad de temperaturas a cargar: '))
    temp = n * [0.0]
    read(temp)

    d = amplitud(temp)

    print('Amplitud térmica:', d)

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

Seleccione una:

- ☒ a. Calcula y retorna la diferencia entre la mayor y la menor temperatura del arreglo *temp*. ✓
- ☐ b. Calcula y retorna la la mayor temperatura del arreglo *temp*.
- ☐ c. Calcula y retorna la menor temperatura del arreglo *temp*.
- ☐ d. Calcula y retorna el promedio entre la mayor y la menor temperatura del arreglo *temp*.

Pregunta 6

Correcta

Se puntúa 1 sobre 1

El siguiente programa crea tres arreglos para cargar y almacenar en ellos los legajos, los nombres y los promedios de n estudiantes. ¿Qué hace finalmente este programa al ejecutarlo?

```
__author__ = 'Cátedra de AED'

def read(leg, nom, pro):
    n = len(leg)
    for i in range(n):
        leg[i] = int(input('Legajo: '))
        nom[i] = input('Nombre: ')
        pro[i] = float(input('Promedio: '))

def search(nom, x):
    n = len(nom)
    for i in range(n):
        if x == nom[i]:
            return i
    return -1

def test():
    n = int(input('Cantidad de alumnos: '))
    leg = n * [0]
    nom = n * ['']
    pro = n * [0.0]
    read(leg, nom, pro)

    x = input('Ingrese el nombre del estudiante a buscar: ')
    ind = search(nom, x)

    if ind != -1:
        print('El estudiante pedido está registrado en la posición', ind, 'y sus datos son:')
        print('Legajo:', leg[ind])
        print('Nombre:', nom[ind])
        print('Promedio:', pro[ind])
    else:
        print('No hay un estudiante con ese nombre...')

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

Seleccione una:

- ☐ a. Busca en el arreglo de nombres un estudiante con nombre igual a x , mediante la función `search()` aplicando búsqueda secuencial. Si tal nombre existe muestra el promedio (y sólo el promedio) de estudiante, y si no existe se muestra el valor *None*.
- ☐ b. Busca en el arreglo de nombres un estudiante con nombre igual a x , mediante la función `search()` aplicando búsqueda secuencial. Si tal nombre existe muestra el índice de la casilla que lo contenía, y si no existe se muestra un mensaje avisando de ese hecho.
- ☒ c. Busca en el arreglo de nombres un estudiante con nombre igual a x , mediante la función `search()` aplicando búsqueda secuencial. Si tal nombre existe muestra todos sus datos del estudiante, y si no existe se muestra un mensaje avisando de ese hecho. ✓
- ☐ d. Busca en el arreglo de nombres un estudiante con nombre igual a x , mediante la función `search()` aplicando búsqueda binaria. Si tal nombre existe muestra todos los datos del estudiante, y si no existe se muestra un mensaje avisando de ese hecho.

Pregunta 7

Correcta

Se puntúa 1 sobre 1

El programa que sigue crea y carga tres arreglos con los legajos, nombres y promedios de n estudiantes. ¿Qué hace concretamente la función **check()** del programa?

```
__author__ = 'Cátedra de AED'

def read(leg, nom, pro):
    n = len(leg)
    for i in range(n):
        leg[i] = int(input('Legajo: '))
        nom[i] = input('Nombre: ')
        pro[i] = float(input('Promedio: '))

def display_all(leg, nom, pro):
    n = len(leg)
    for i in range(n):
        print('Legajo:', leg[i], end=' ')
        print('Nombre:', nom[i], end=' ')
        print('Promedio:', pro[i])

def check(pro):
    for p in pro:
        if p < 4:
            return True
    return False

def test():
    n = int(input('Cantidad de alumnos: '))
    leg = n * [0]
    nom = n * ['']
    pro = n * [0.0]
    read(leg, nom, pro)

    print()
    display_all(leg, nom, pro)

    if check(pro):
        print('Se ha registrado al mnenos un estudiante aplazado...')
    else:
        print('No se han registrado estudiantes aplazados...')

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

Seleccione una:

- ☐ a. Retorna la suma acumulada de todos los promedios del arreglo *pro*.
- ☐ b. Retorna *True* si todos los valores del vector *pro* son mayores o iguales a 4, y *False* si al menos un promedio es menor a 4.
- ☐ c. Retorna *True* si el arreglo *pro* contiene alguna casilla valiendo *None*, y retorna *False* si todos los casilleros son diferentes de *None*.
- ☒ d. Retorna *True* si el arreglo *pro* contiene al menos un valor menor a 4, y retorna *False* si todos los promedios del arreglo son mayores o iguales a 4. ✓

Pregunta 8

Correcta

Se puntúa 1 sobre 1

¿Qué hace la siguiente función en Python?

```
def contar(n):  
    v = n * [0]  
  
    num = int(input('Ingrese un valor entre 0 y' + str(n) + '(con -1 corta):'))  
    while num != -1:  
        if 0 <= num < n:  
            v[num] += 1  
        else:  
            print('Error. El número debe ser >= 0 y <', n)  
            num = int(input('Ingrese otro valor entre 0 y' + str(n) + '(con -1 corta):'))  
  
    return v
```

Seleccione una:

- ☒ a. Carga por teclado una secuencia de números, y usa el vector **v** para contar cuántas veces apareció cada número. ✓
- ☐ b. Carga por teclado una secuencia de números, y usa el vector **v** para almacenar esos números.
- ☐ c. Carga por teclado una secuencia de números, y usa el vector **v** para acumular esos números.
- ☐ d. Carga por teclado una secuencia de números, y usa el vector **v** para validar que esos números estén dentro del rango [0, n-1].

Pregunta 9

Correcta

Se puntúa 2 sobre 2

Suponga que la variable *codigos* tomada como parámetro en la siguiente función se usa para almacenar números entre 0 y 24 que indican en qué idioma están escritos los *n* libros de un conjunto de libros que se tiene que procesar (por ejemplo, 0: español, 1: inglés, 2: italiano, etc.) Asuma que ese arreglo de códigos fue creado y cargado correctamente antes de ser invocada la función. ¿Qué hace entonces la siguiente función?

```
def procesar(codigos):  
    n = len(codigos)  
    c = 25 * [0]  
    for i in range(n):  
        id = codigos[i]  
        if 0 <= id <= 24:  
            c[id] += 1  
  
    return c
```

Seleccione una:

- ☐ a. Usa un vector *c* de números enteros, para contar cuántos libros hay en cada idioma, y luego retorna la conversión a cadena de caracteres del vector de conteos *c*.
- ☒ b. Usa un vector *c* de números enteros, para contar cuántos libros hay en cada idioma, y luego retorna el vector de conteos *c*. ✓
- ☐ c. Usa un vector *c* de números enteros, para contar cuántos libros hay en cada idioma, y luego retorna el valor más alto del vector de conteos.
- ☐ d. Usa un vector *c* de números enteros, para contar cuántos libros hay en cada idioma, y luego retorna el vector original *codigos*.

Pregunta **10**

Correcta

Se puntúa 2 sobre 2

El siguiente programa crea y carga dos arreglos paralelos *destinos* y *montos* con los datos de n llamadas telefónicas registradas para un cliente en distintos momentos. En el arreglo *destinos* se almacenan los códigos de los lugares de destino de cada llamada y se asumen que esos destinos se representan con números entre 0 y 24 (por ejemplo: 0: Estados Unidos, 1: Brasil, etc.) El arreglo *montos* almacena el costo de cada llamada. ¿Qué hace exactamente el programa completo al ejecutarse?

```
__author__ = 'Cátedra de AED'

def read(destinos, montos):
    n = len(destinos)
    for i in range(n):
        destinos[i] = int(input('Código de destino de la llamada (valor entre 0 y 24 por favor): '))
        montos[i] = float(input('Monto: '))

def process(destinos, montos):
    n = len(destinos)

    s = 25 * [0]
    for i in range(n):
        d = destinos[i]
        if 0 <= d <= 24:
            s[d] += montos[i]

    return s

def display(s):
    print('Listado solicitado de llamadas...')

    m = len(s)
    for i in range(m):
        if s[i] != 0:
            print('Destino:', i, 'Total:', s[i])

def test():
    n = int(input('Cantidad de llamadas: '))
    destinos = n * [0]
    montos = n * [0.0]
    read(destinos, montos)

    s = process(destinos, montos)

    print()
    display(s)

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

Seleccione una:

- ☐ a. Muestra los montos acumulados en llamadas realizadas para cada uno de los 25 posibles destinos (incluyendo en el listado los destinos cuyo monto acumulado sea cero).
- ☐ b. Muestra una línea única con el monto total acumulado entre todas las llamadas realizadas.
- ☐ c. Muestra la cantidad de llamadas realizadas para cada uno de los 25 posibles destinos (sólo para los destinos cuyo cantidad de llamadas sea diferente de cero).
- ☒ d. Muestra los montos acumulados en llamadas realizadas para cada uno de los 25 posibles destinos (sólo para los destinos cuyo monto acumulado sea diferente de cero) ✓

[◀ Materiales Adicionales para la Ficha 17](#)

Ir a...



