Guía de Ejercicios Prácticos - Ficha 14

Sitio: Universidad Virtual UTN FRC
Curso: Algoritmos y Estructuras de Datos (2020)
Libro: Guía de Ejercicios Prácticos - Ficha 14
Imprimido por: Luciana Lisette Montarce
Día: lunes, 23 de noviembre de 2020, 21:24

Descripción

Esta guía contiene enunciados de algunos ejercicios para aplicar los conceptos de programación en ***Python*** que se analizan en la ***Ficha 14***. Los alumnos no deben subir nada al aula virtual: la guía se propone como fuente de ejercicios generales. Se sugiere intentar resolver cada uno de estos problemas ya sea trabajando solos o en grupos de estudio, y cuando las soluciones se publiquen, controlar lo hecho con las sugerencias propuestas por sus docentes. Utilice el foro del curso para plantear dudas y consultas, que cualquier alumno puede intentar responder.

Tabla de contenidos

1. Estudio climatológico
1.1. Solución

2. Estadística con dados
2.1. Solucion

3. Ventas
3.1. Solución

4. Repaso de arreglos
4.1. Solución

1. Estudio climatológico

Como parte de un estudio climatológico, se desea un programa que permita obtener una serie de estadísticas a partir de un conjunto de muestras de temperatura.

Se pide un programa que:

- Ingrese n muestras de temperatura, donde cada muestra contiene la temperatura registrada, la región donde se registró la misma (1-20), y el día del mes en el que se registró la temperatura
- Determinar el promedio general de temperatura
- Dada una región, mostrar las temperaturas de la misma, ordenadas por dia, de menor a mayor
- Dada una región, determinar si la temperatura de alguna muestra superó el valor x, ingresado por teclado.
- Determinar la cantidad de muestras por region (20 contadores)

1.1. Solución

```
def cargar_datos():
    n = int(input("Ingrese la cantidad de muestras a registrar: "))
    temperaturas = [0] * n
    regiones = [0] * n
    dias = [0] * n
    for i in range(n):
        dia = int(input("Ingrese día: "))
        region = int(input("Ingrese región: "))
        temperatura = float(input("Ingrese temperatura: "))
        dias[i] = dia
        regiones[i] = region
        temperaturas[i] = temperatura
    return dias, regiones, temperaturas


def menu():
    print("1 _ Cargar datos")
    print("2 _ Promedio de temperaturas")
    print("3 _ Mostrar temperatura de una región")
    print("4 _ Buscar temperaturas mayores a x")
    print("5 _ Salir")
    return int(input("Ingrese opción: "))


def promedio_temperaturas(temperaturas):
    suma = 0
    for temp in temperaturas:
        suma += temp
    return suma / len(temperaturas)


def buscar_temperatura(regiones, temperaturas, region, x):
    existe = False
    for i in range(len(regiones)):
        if regiones[i] == region and temperaturas[i] > x:
            existe = True
            break

    return existe


def ordenar(dias, regiones, temperaturas):
    n = len(dias)
    for i in range(n - 1):
        for j in range(i + 1, n):
            if dias[i] > dias[j]:
                dias[i], dias[j] = dias[j], dias[i]
                regiones[i], regiones[j] = regiones[j], regiones[i]
                temperaturas[i], temperaturas[j] = temperaturas[j], temperaturas[i]


def mostrar_temperaturas(dias, regiones, temperaturas, reg):
    print("Dia \t\t Temperatura")
    for i in range(len(regiones)):
        if regiones[i] == reg:
            print(dias[i], "\t\t\t", temperaturas[i])


def main():
    op = 0
    while op != 5:
        op = menu()
        if op == 1:
            dias, regiones, temperaturas = cargar_datos()
        elif op == 2:
            promedio = promedio_temperaturas(temperaturas)
            print("El promedio de temperaturas fue de: ", promedio)
```

```
        elif op == 3:
            ordenar(dias, regiones, temperaturas)
            reg = int(input("Ingrese región a analizar: "))
            mostrar_temperaturas(dias, regiones, temperaturas, reg)
        elif op == 4:
            reg = int(input("Ingrese región a analizar: "))
            x = int(input("Ingrese temperatura a controlar: "))
            existe = buscar_temperatura(regiones, temperaturas, reg, x)
            if existe:
                resultado = "Hay al menos una temperatura menor a"
            else:
                resultado = "No hay temperaturas menores a"
            print(resultado, x, "en la región analizada")
        elif op == 5:
            print("Hasta luego.")
        else:
            print("Error, opción errónea")


if __name__ == "__main__":
    main()
```

2. Estadística con dados

Para realizar una prueba estadística, se lanzan simultáneamente 2 dados y se anotan los resultados obtenidos. Al cabo de n lanzamientos, se necesita determinar:

- Cuántas veces se obtuvo el mismo valor en ambos dados y qué porcentaje representa sobre el total de lanzamientos.
- En qué lanzamiento se dio por primera vez una suma impar entre ambos dados
- Cuál fue el mayor valor que apareció en cada dado y cuántas veces se presentó
- Cuántas veces apareció cada una de las sumas posibles entre ambos dados. Es decir: cuántas veces sumaron 2, cuántas veces sumaron 3, y así sucesivamente
- Determinar la cantidad de tiradas en las que la suma de ambos dados fue mayor que la suma promedio de todas las tiradas

Desarrollar un programa que permita simular esta situación, guardando en vectores paralelos los valores de ambos dados.

2.1. Solucion

```
__author__ = 'Cátedra de Algoritmos y Estructuras de Datos'

def validar_tamanio():
    n = int(input('Ingrese el tamaño del arreglo: '))
    while n<= 0:
        n = int(input('Debe ser un valor positivo. Ingrese otro: '))
    return n


def validar_dado(mensaje):
    dado = int(input(mensaje))
    while dado <1 or dado > 6:
        dado = int(input('Error!' + mensaje))
    return dado

def cargar(n):
    dado1 = list()
    dado2 = list()
    for i in range(n):
        print('Lanzamiento',i+1)
        dado = validar_dado('Ingrese el valor del dado 1: ')
        dado1.append(dado)
        dado = validar_dado('Ingrese el valor del dado 2: ')
        dado2.append(dado)
    return dado1, dado2


def contar_iguales(dado1, dado2):
    iguales = 0
    for i in range(len(dado1)):
        if dado1[i] == dado2[i]:
            iguales += 1
    return iguales


def buscar_suma_impar(dado1, dado2):
    for i in range(len(dado1)):
        if (dado1[i]  + dado2[i]) % 2 != 0:
            return i+1
    return -1


def buscar_mayor(v):
    mayor = v[0]
    repeticiones = 1
    for i in range(1,len(v)):
        if v[i] > mayor:
            mayor = v[i]
            repeticiones = 1
        elif v[i] == mayor:
            repeticiones += 1
    return mayor, repeticiones


def contar_sumas(dado1, dado2):
    conteo = [0]*13
    for i in range(len(dado1)):
        suma = dado1[i] + dado2 [i]
        conteo[suma] += 1
    return conteo


def mostrar(conteo):
    for i in range(len(conteo)):
        print('Suma',i,'->',conteo[i],'veces')

def sumas_mayores_promedio(dado1, dado2):
```

```
    ac = 0
    for i in range(len(dado1)):
        ac += dado1[i]
        ac += dado2[i]

    promedio = ac / len(dado1)
    cant_mayores = 0

    for i in range(len(dado1)):
        s = dado1[i] + dado2[i]
        if s > promedio:
            cant_mayores += 1
    return cant_mayores


def test():
    print('PROGRAMA ESTADÍSTICO')
    n = validar_tamanio()
    dado1, dado2 = cargar(n)
    print()
    print('Dado 1', dado1)
    print('Dado 2', dado2)
    print()
    iguales = contar_iguales(dado1,dado2)
    porcentaje = round(iguales * 100 / n, 2)
    print('Los dados fueron iguales',iguales,'veces y representa el',porcentaje,'% del total')
    print()
    lanzamiento = buscar_suma_impar(dado1, dado2)
    if lanzamiento == -1:
        print('Nunca hubo una suma impar')
    else:
        print('La primer suma impar apareció en el lanzamiento',lanzamiento)
    print()
    mayor, repeticiones = buscar_mayor(dado1)
    print('El mayor valor del dado 1 es',mayor,'y apareció',repeticiones,'veces')
    mayor, repeticiones = buscar_mayor(dado2)
    print('El mayor valor del dado 2 es',mayor,'y apareció',repeticiones,'veces')
    conteo = contar_sumas(dado1,dado)
    print()
    print('Repeticiones')
    mostrar(conteo)
    cantidad_mayores = sumas_mayores_promedio(dado1, dado2)
    print('En', cantidad_mayores, 'tiradas la suma fue mayor al promedio')

test()
```

3. Ventas

Se pide cargar las n ventas del día de un comercio en 2 vectores. De cada venta se conoce el artículo adquirido (son 4 tipos de artículos numerados del 0 al 3) y la cantidad vendida de dicho artículo en esa venta.

Se debe calcular y mostrar la cantidad vendida de cada uno de los 4 artículos.

3.1. Solución

```
__author__ = 'Algoritmos y Estructuras de Datos'

def validate(mensaje):
    n = 0
    while n <= 0:
        n = int(input(mensaje))
        if n <= 0:
            print("Valor incorrecto!")
    return n

def validate_range(inf, sup):
    n = inf - 1
    while n < inf or n > sup:
        n = int(input("Ingrese el tipo de artículo, entre "+ str(inf) + " y " + str(sup) + ": "))
        if n < inf or n > sup:
            print("Valor incorrecto!")
    return n

def read(art, cant):
    for i in range(len(art)):
        print("Venta", i)
        art[i] = validate_range(0, 3)
        cant[i] = validate("Ingrese la cantidad vendida de dicho articulo: ")

def write(art, cant):
    for i in range(len(art)):
        print("Venta:", i)
        print("Artículo:", art[i], "- Cantidad:", cant[i])

def contar(art, cant):
    acu = [0] * 4
    for i in range(len(art)):
        acu[art[i]] += cant[i]

    print("\nCantidades vendidas por artículo")
    for i in range(len(acu)):
        print("Cantidad vendida del articulo", i, ":", acu[i])

def test():
    n = validate("Ingrese la cantidad de ventas del día: ")
    art = [0] * n
    cant = [0] * n
    read(art, cant)
    write(art, cant)
    contar(art, cant)


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

4. Repaso de arreglos

Generar un programa que cargue un lote de números enteros y positivos y que a través de un menú de opciones le permita al usuario

1. Obtener el promedio de los números comprendidos entre dos valores ingresados por el usuario
2. Obtener el menor número impar del lote
3. Imprimir todos los números múltiplos de un valor ingresado por el usuario separados por comas

4.1. Solución

```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'
import random


def mayor_cero():
    numero = 0
    while numero <= 0:
        numero = int(input('Ingrese un numero: '))
        if numero <= 0:
            print('Valor Incorrecto!! Debe ser superior a cero')
    return numero

def generar_lote_numero(cantidad):
    lote = []

    for i in range(cantidad):
        lote.append(random.randint(1, 150))
    return lote


def calcular_promedio(lote, inferior, superior):
    acumulador = 0
    contador = 0
    for pos in range(len(lote)):
        if inferior <= lote[pos] <= superior:
            contador += 1
            acumulador += lote[pos]

    if contador > 0:
        return acumulador / contador
    else:
        return 0


def listar_multiplos(lote, multiplo):
    res = ''
    for pos in range(len(lote)):
        if lote[pos] % multiplo == 0:
            res += str(lote[pos]) + ', '
    return res


def menor_numero_impar(lote):
    menor = 0
    primero = True
    for pos in range(len(lote)):
        if lote[pos] % 2 != 0:
            if primero:
                menor = lote[pos]
                primero = False
            elif menor > lote[pos]:
                menor = lote[pos]
    return menor


def test():
    menu = 'Menu de opciones \n' \
           '================================\n' \
           '1 \t Promedio ente dos valores \n' \
           '2 \t Menor numero impar \n' \
           '3 \t Imprimir múltiplos \n' \
           '4 \t Salir \n' \
           'Ingrese su opcion: '
    cantidad = mayor_cero()
    lote = generar_lote_numero(cantidad)
    opcion = 0
    while opcion != 4:
```

```
        opcion = int(input(menu))
        if opcion == 1:
            valor1 = mayor_cero()
            valor2 = mayor_cero()

            if valor1 > valor2:
                inferior, superior = valor2, valor1
            else:
                inferior, superior = valor1, valor2

            promedio = calcular_promedio(lote, inferior, superior)
            print('El promedio de numeros es ', promedio)

        elif opcion == 2:
            print("El menor numero impar es ", menor_numero_impar(lote))
        elif opcion == 3:
            multiplo = mayor_cero()
            print(listar_multiplos(lote, multiplo))


test()
```