

Guía de Ejercicios Prácticos 20

Sitio: [Universidad Virtual UTN FRC](https://uv.frc.utn.edu.ar)
Curso: Algoritmos y Estructuras de Datos (2020)
Libro: Guía de Ejercicios Prácticos 20

Imprimido por: Luciana Lisette Montarce
Día: lunes, 23 de noviembre de 2020, 21:29



Descripción

Esta guía contiene enunciados de algunos ejercicios para aplicar los conceptos de programación en **Python** que se analizan en la **Ficha 20**. 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. Carreras

2. Liga de Futbol Americano

3. US Open

4. Empresa celulares

5. Empresa de TV+Internet

5.1. principal_base

5.2. registros_base



FILADD.COM

1. Carreras

Se tienen los resultados de una carrera de autos. De cada participante se conoce: nombre, tiempo (segundos) y cantidad de participaciones previas (se sabe que el corredor que más participaciones tiene es 8).

- 1) Cargar el vector de n corredores, siendo n un valor ingresado por teclado.
- 2) Mostrar los corredores con más de p participaciones, siendo p cargado por teclado.
- 3) Determinar la cantidad de corredores por cantidad de participaciones.
- 4) Mostrar los corredores ordenados por el tiempo.
- 5) Buscar en el vector un corredor con nombre x . Si existe mostrar sus datos, caso contrario indicar un mensaje de error.



FILADD.COM

2. Liga de Futbol Americano

La liga amateur de Futbol Americano de Argentina necesita un programa para obtener algunas estadísticas de los jugadores que participan. De cada jugador se conoce:

- Nombre
- Numero de camiseta (un número entre 0 y 99)
- Numero de equipo (los 18 equipos participantes estan numerados correlativamente a partir de 1)
- Cantidad de yardas recorridas (un número entero)
- Cantidad de puntos marcados (un número entero)

A partir de los registro cargados resolver:

- 1) Cargar un vector con n jugadores cuyos datos se ingresan por teclado.
- 1.Bis) Generar datos de jugadores en un vector con valores aleatorios
- 2) Mostrar los jugadores ordenados por nombre
- 3) Determinar la cantidad de jugadores con yardas recorridas mayores que las yardas promedio de todos los jugadores.
- 4) Buscar el jugador con la camiseta x del equipo y donde x y y se cargan por teclado y, en el caso de encontrarlo solicitar una cantidad de yardas y añadirlas a las yardas recorridas por el jugador. Si no se encuentra mostrar un mensaje.
- 5) Promedio de yardas recorridas por equipo
- 6) Generar y mostrar un nuevo vector con los jugadores que convirtieron puntos.



3. US Open

La administración del US Open (Abierto de Tenis de los estados unidos) necesita un programa para mantener la estadística de los jugadores que participan en el torneo, de cada jugador pretende almacenar: el número de ingreso el cuadro (un número único de 1 en adelante para cada jugador), el Nombre del jugador, el origen de entrada al torneo (un número de 0 a 3 que significa 0 - Preclasificado, 1 - Wildcard, 2 - Qualy), la cantidad de sets perdidos en lo que va del torneo, el tiempo total jugado en lo que va del torneo y si aún está en juego o ya perdió.

De acuerdo con los datos pedidos se pide un programa con menú de opciones que mantenga un vector con todos los jugadores cargados y permita las siguientes opciones:

1. Agregar n jugadores donde n se carga por teclado, validando que el nro de ingreso sea único y el origen esté efectivamente entre 0 y 3.
2. Permitir agregar un jugador más al conjunto de jugadores cargados con las mismas validaciones del punto anterior.
3. Buscar un jugador con un número x que se carga por teclado, si lo encuentra mostrar sus datos y permitir modificar la cantidad de sets perdidos, el tiempo total de juego y si sigue en juego o no. Si no lo encuentra mostrar un mensaje de error.
4. Determinar la cantidad de jugadores con tiempo de juego mayor al tiempo promedio, solo de los jugadores que aún están en carrera.
5. Determinar la cantidad de jugadores por origen mostrando un listado con la descripción del origen y la cantidad para el mismo. Usar vector de conteo.
6. Mostrar los jugadores que aún siguen en el torneo ordenados de menor a mayor por tiempo de juego.

4. Empresa celulares

Una empresa dedicada a la venta de líneas para celulares nos pidió un programa que permita realizar una serie de informes. De cada línea se sabe número, titular, tipo de producto (valor de 0 a 19), cantidad de minutos consumidos

Usted debe realizar dicho programa, controlado por un menú de opciones para que lleve a cabo los siguientes ítems:

1. Cargar un vector con n Líneas, validando que el tamaño a cargar sea mayor a cero y el tipo de producto sea válido
2. Listar todas las líneas ordenadas por minutos consumidos en forma descendente
3. Determinar la cantidad de líneas que superan un valor X ingresado por teclado de minutos consumidos y que porcentaje representa sobre el total de líneas del arreglo
4. Determinar e informar la cantidad de minutos consumidos por cada tipo de plan (son 20 contadores por cada tipo de plan)
5. Mostrar la línea con menor cantidad de minutos consumidos, en caso que haya más de una mostrarlas a todas
6. Buscar una línea X ingresada por teclado, para un tipo de producto Y también ingresado por teclado. Incrementar sus minutos consumidos en un 20% y mostrar los datos de la línea. Caso contrario indicar con un mensaje que no existe



5. Empresa de TV+Internet

Una empresa proveedora de servicios de TV e Internet solicita un programa para gestionar su facturación.

Por cada cliente se define: identificación (un valor único), nombre del titular, zona geográfica (un valor entre 0 y 9 inclusive), tipo de cliente (0: Residencial / 1: Edificio / 2: Empresa), abono TV básico, abono TV Premium, abono internet.

Se solicita cargar un vector de clientes y, a partir del vector, generar una tabla que contenga la cantidad de clientes por zona y tipo.

Luego, implementar un menú de opciones que permita:

1. Mostrar el contenido del vector.
2. Generar un nuevo vector de facturación, sólo para clientes residenciales. Cada registro debe contener: identificación, titular, cantidad de servicios contratados, monto total facturado.
3. Mostrar cantidad de clientes por zona y tipo (sólo valores mayores a 0).
4. Calcular total de clientes, para una zona ingresada por teclado.
5. Informar total de clientes, para un tipo que se ingresa por teclado, y qué porcentaje representa sobre el total de clientes.



5.1. principal_base



```
import random
from registros import *

def validar_mayor_que(valor, mensaje):
    num = int(input(mensaje))
    while num <= valor:
        print('Valor invalido')
        num = int(input(mensaje))
    return num

def validar_entre(desde, hasta, mensaje):
    num = int(input(mensaje))
    while num < desde or num > hasta:
        print('Valor invalido')
        num = int(input(mensaje))
    return num

def cargar_automatico(v, n):
    for i in range(n):
        id = 10 + i * 2
        titular = 'Cliente ' + str(i)
        zona = random.randint(0, 9)
        tipo = random.randint(0, 2)
        tvbasico = random.choice((0, 1250, 1500, 1720))
        if tvbasico == 0:
            tvpremium = 0
        else:
            tvpremium = random.choice((0, 2000))
            internet = random.choice((0, 1300, 2400))
        v.append(Cliente(id, titular, zona, tipo, tvbasico, tvpremium, internet))

def mostrar_vector(v):
    for i in range(len(v)):
        print(to_string(v[i]))

def mostrar_menu():
    print('\nSERVICIOS DE TV E INTERNET')
    print('1. Listado de clientes')
    print('2. Facturación clientes residenciales')
    print('3. Clientes por zona y tipo')
    print('4. Total de clientes de una zona')
    print('5. Cantidad y porcentaje de clientes de un tipo')
    print('0. Salir')
    opcion = int(input('Ingrese su opcion: '))
    return opcion

def principal():
    v = list()
    n = validar_mayor_que(0, 'Ingrese cantidad de clientes: ')
    cargar_automatico(v, n)
    opcion = -1
    while opcion != 0:
        opcion = mostrar_menu()
        if opcion == 1:
            mostrar_vector(v)
        elif opcion == 0:
            print('Hasta pronto!')

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



5.2. registros_base

```
class Cliente:
    def __init__(self, id, titular, zona, tipo, tvbasico, tvpremium, internet):
        self.id = id
        self.titular = titular
        self.zona = zona
        self.tipo = tipo
        self.tvbasico = tvbasico
        self.tvpremium = tvpremium
        self.internet = internet

def to_string(reg):
    cad = 'Id: {:<4} - Titular: {:<10} - Zona: {} - Tipo: {} - TV Basico $ {:>4} - TV Premium $ {:>4} - Internet $ {:>4}'
    return cad.format(reg.id, reg.titular, reg.zona, reg.tipo, reg.tvbasico, reg.tvpremium, reg.internet)
```