IES SAN JUAN DE LA RAMBLA
DEPARTAMENTO DE INFORMÁTICA Y COMUNICACIONES
CFGS DESARROLLO DE APLICACIONES MULTIPLATAFORMA
MÓDULO: PROGRAMACIÓN

San Juan de la Rambla a 02 de marzo de 2022

UT03- Prueba práctica – RECUPERACIÓN

Esta prueba pertenece a la evaluación de la UT03 y pertenece a la parte que vale un 75% de la nota en el procedimiento de recuperación.

Esta parte de desarrollo tiene un valor de 10 puntos. Aquí si pueden utilizar el material que quieran.

NO SE PERMITE EL USO DE INSTRUCCIONES QUE NO SE HAYAN TRABAJADO EN CLASE, si la solución que presenten es buscada en internet NO SERÁN CORREGIDAS.

Forma de entrega, se debe enviar los ficheros .py de los problemas que resuelvan a EVAGD en el plazo determinado para ello. Entregas fuera de plazo, o por otros medios NO SERÁN CORREGIDAS.

Problema 01: (3,0 pts.)

La probabilidad de que salga un número al tirar un dado es de 1/6 = 0,1666….7.

Un experimento para comprobar lo que dice la teoría, sería lanzar un número grande de veces un dado, supongamos 20.000, y anotar en cada tirada el número que sale. Una vez realizados todos los lanzamientos, se cuenta el número de veces que salió cada número del 1 al 6 y se divide entre el número total de tiradas. Así se puede obtener un cálculo aproximado del valor de la probabilidad para cada número del dado.

Lo que se debe realizar.

Un programa en Python que realice las 20.000 tiradas del dado usando la función *random.random()* de la librería *random*.

La función *random()* genera números aleatorios con valores entre $0 \leq$ número < 1. Es decir, valores entre 0 y 1, por ejemplo, 0,97, no tomando nunca el valor de 1. Por lo que deberán trabajar con el número obtenido para que genere un número entre 1 y 6 como los que salen en el dado.

Para llevar la cuenta del número de veces que sale un número se debe implementar un diccionario en el que cada elemento será de la siguiente forma: **clave:valor**

Donde la **clave** será cada una de las caras del dado, o sea, los números del 1 al 6, y el **valor** será el número de veces que ese número ha salido.

El resultado final que debe mostrar el programa es el cálculo de la probabilidad obtenida mediante la simulación del experimento para cada una de las caras del dado y la diferencia que hay con el valor real de la misma.

Problema 02: (4,0 pts.)

Se quiere mantener una agenda mediante un diccionario. Cada elemento de ese diccionario tendrá como clave una fecha **"dd/mm/aa"** y como valor una lista de actividades a realizar ese día.

Cada elemento de esa lista será una **tupla** con tres valores:

(hora_inicio (hh:mm), duración (hh:mm), actividad)

El programa principal debe presentar el siguiente menú:

1. Agendar actividad
2. Borrar actividad
3. Listar actividades
4. Salir

Lo que debe realizar cada opción:

1. Debe añadir una actividad en la agenda (siempre que no exista otra con la misma fecha y hora). En caso de que ya exista una cita, se debe indicar mediante un mensaje de error. Se deben solicitar al usuario todos los datos necesarios para concretar la cita.
2. Debe borrar una actividad de la agenda (siempre que exista). En caso de que NO exista se debe mostrar el correspondiente mensaje de error. Para borrar una cita se debe solicitar al usuario el día y la hora de la misma.
3. Debe listar las actividades para un día que se le solicita al usuario. Si el dato introducido no tiene citas registradas debe mostrar el siguiente mensaje: "No hay citas programadas para esa fecha". En caso de que la fecha si esté agendada, se debe mostrar todas las actividades una por cada fila mostrado la hora, duración y la actividad.
4. Debe terminar la ejecución del programa.

Ejercicio 03: (3,0 puntos)

Se quiere crear un programa que tenga el siguiente menú para trabajar con una lista de números enteros.

1) Añadir número a la lista: Pide un número y lo añade al final de la lista.
2) Añadir número de la lista en una posición: Pide un número y una posición, si la posición existe, es decir, es menor que el número de elementos que hay en la lista, se inserta el número en esa posición, desplazando el resto de elementos una posición hacia adelante.
3) Eliminar un número: Borra todas las apariciones del número de la lista.
4) Eliminar un número de una posición: Pide un número y una posición, si la posición existe, es decir, es menor o igual que el número de elementos que hay en la lista, se borra el número en esa posición.
5) Contar números: Te pide un número y te dice cuantas veces aparece en la lista.
6) Posiciones de un número: Te pide un número y te dice todas las posiciones en las que encuentra.
7) Mostrar números: Muestra los números de la lista (sin repetir)
8) Muestra estadísticas: Muestra el siguiente submenú:
   a) Cantidad de números con 1, 2 y 3 cifras respectivamente.
   b) Cantidad de números pares e impares.
   c) Suma de todos los números
   d) Regresar
9) Salir

Cosas a tener en cuenta:

- Los números a almacenar van del 1 al 999 ambos inclusive.
- Los números pueden estar repetidos en la lista.
- La posición 0 de la lista tendrá un significado especial, en esa posición siempre se debe tener actualizado el valor del número de elementos que hay en la lista.
- Debido a lo anterior, los números empiezan a almacenarse a partir de la posición 1. Se debe tener en cuenta este dato para cuando se tenga que borrar o insertar un número en una posición dada, la posición que se le pide al usuario será la posición real del elemento en la lista.
- Cada vez que se realice una operación de añadir o eliminar números de la lista se debe al finalizar, actualizar el número de elementos de la lista.