**EJERCICIOS CICLOS SIMPLES, ANIDADOS, ABTRACCIÓN**

**Tipo parcial**

1. Usted debe desarrollar una operación que dados **N** equipos de futbol de una liga, donde cada equipo juega **P** partidos, solicite por teclado: el **nombre del equipo**, y por cada partido (No se requiere el rival): los goles a favor y los goles en contra. El procedimiento debe responder a las siguientes preguntas:

¿Cuántos puntos obtuvo cada equipo en la liga? (si gana obtiene 3 puntos, si empata 1 punto. y si pierde 0 puntos.)

¿Cuál fue la diferencia de goles de cada equipo? (total goles a favor - total goles en contra del equipo)

¿Cuál es el nombre del equipo que ganó la liga (obtuvo el mayor número de puntos)?

1. Los entrenadores de 15 atletas han descubierto una rutina que permite a la persona incrementar su rendimiento en un periodo de 30 días. Cuando un deportista entra en el programa de alto rendimiento, el entrenador solicita tres datos: el peso en kilogramos, la estatura en metros y la velocidad promedio en metros/minuto.

La fórmula indica que el deportista debe realizar un número de vueltas N, de distancia D, que se determinan con la siguiente fórmula:

* Si el número del día corresponde con un número primo, la fórmula es: N= 10;
* Si el número del día no corresponde con un número primo:

N= (Nant + ((estatura + número del día actual) \* velocidad) / peso)

Nant = Número de vueltas del día anterior

* La distancia D= (330 metros \* estatura)

Cree la función **primo**(N) que valida si el día N, es o no un numero primo, retornando True o False según sea el caso. Defina una operación, que invocando la función **primo**(N), imprima las siguientes 3 cosas: (i) el número de vueltas que debe realizar cada deportista por día y además imprima al final: (ii) el total de vueltas que realizó la persona durante los 30 días que dura el programa y (iii) la distancia total recorrida en metros.

1. Un grupo de jóvenes Javerianos, que participan del voluntariado social universitario que dirige el Centro Pastoral San Francisco Javier, apoya el proyecto Escuela de Liderazgo Manresa mediante el recaudo de recursos. El coordinador del proyecto sabe que cada semestre cuenta con un número *(n)* de jóvenes voluntarios; cada joven tiene la oportunidad de diseñar *(x)* estrategias para recaudar fondos durante el semestre. Cuando el joven plantea una estrategia, el mismo joven se pone un tope o meta en pesos para recaudar con cada estrategia. El joven debe tener dos estrategias como mínimo.

El éxito o fracaso de cada estrategia depende de la creatividad, la experiencia, la carrera y el empeño del joven. El coordinador del proyecto discrimina las estrategias de acuerdo con los siguientes parámetros: si la estrategia logra captar más del 90% del tope propuesto, la estrategia se discrimina como una experiencia exitosa; si la estrategia permite captar entre el 50% y 90% del tope propuesto, la estrategia se discrimina como una estrategia adecuada; y si la estrategia capta menos del 50% del tope propuesto, la estrategia es discriminada como inadecuada.

Al principio del semestre, el coordinador del proyecto pide a un estudiante de Introducción a la Programación que le ayude con la elaboración de un programa en Python, que mediante un procedimiento **voluntarios(n)**, modele un escenario de los resultados del proyecto de recaudo. Para ello el coordinador ofrece los siguientes datos:

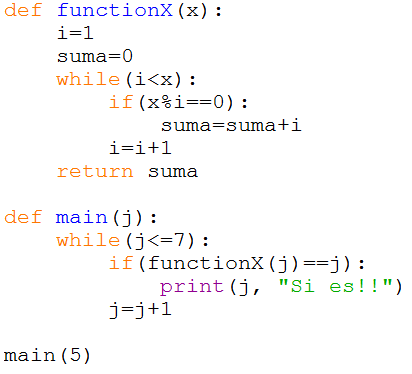
Número de estrategias *(x)* por cada voluntario y el valor en pesos del tope que cada voluntario asume por estrategia. Para validar la estrategia de recaudo se debe utilizar la función randint() (esta función toma como parámetros dos enteros y retorna otro número entero al azar entre dichos dos enteros). El valor arrojado por la función aplicada con los números 0 y 100, dividido entre 100, debe multiplicarse por el valor del tope y este resultado sería el dinero que recauda el estudiante por estrategia. Al final el programa debe imprimir:

* + El recaudo completo por cada voluntario.
  + El recaudo total del semestre
  + El número total de estrategias discriminadas en exitosas.

1. Observe con atención el siguiente código en Python, él está conformado por una operación **functionX** y una función **main** adicional a el llamado a la función main con la cual comienza la ejecución del programa.

Su tarea es describir TODOS los cambios de estado, desde el primera llamado al procedimiento main, hasta que finalice la ejecución del programa.

Tenga en cuenta que describir los cambios de estado, es similar a hacer la prueba de escritorio del programa, es decir, ir analizando paso por paso el valor que va tomando cada una de las variables en cada cambio de estado.



Para describir los cambios de estado, se puede apoyar en una tabla como la siguiente, en la cual va registrando cada uno de los valores de la variable. Si necesita crear más tablas para llevar el registro de forma más clara, lo puede hacer

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nombre variable** | **valor** | **valor** | **valor** | **valor** | **valor** | **valor** | **valor** | **valor** |
| j |  |  |  |  |  |  |  |  |
| i |  |  |  |  |  |  |  |  |
| suma |  |  |  |  |  |  |  |  |
| x |  |  |  |  |  |  |  |  |

1. La Familia Díaz De la Calle desea emprender en vacaciones un viaje por toda Colombia, pero desea calcular cuantas ciudades puede visitar con una cantidad de dinero dada. Se deben tener en cuenta las siguientes consideraciones:

* El vehículo de la familia puede desplazarse hasta 390 Km con un tanque lleno de gasolina. El llenar el tanque de gasolina tiene un costo de $130.000.
* Cada día de viaje la familia gasta $350.000 en alimentación y hoteles.
* Cuando la familia tanquea el carro, le informan en la gasolinera cuantas ciudades se encuentran en los próximos 390 Km.
* Se asume que cada día la familia puede visitar la tercera parte de las ciudades que se encuentran en los próximos 390 Km.
* Se asume que el día de inicio del viaje el vehículo se debe tanquear full.
* Se asume que siempre se tanquea a primera hora de la mañana.

Con la información anterior se debe realizar un programa en Python que:

1. Pregunte por teclado el dinero que tiene disponible la familia para viajar.
2. Imprima, para cada día del viaje, cuantos Km se han recorrido hasta el momento, cuantas ciudades se han visitado y cuánto dinero queda disponible (al final de la jornada y después de pagar los gastos del día).
3. Si al llegar a la gasolinera se detecta que el dinero sobrante no alcanza para tanquear, se deberá detener el programa, informando cuantas ciudades se visitaron y cuantos Km se recorrieron.
4. Si en el transcurso del viaje se detecta que el dinero sobrante no alcanza para la alimentación y hospedaje del día, se deberá detener el programa, informando cuantas ciudades se visitaron y cuantos Km se recorrieron.

El resultado del programa debería ser algo similar a:

