

▼ Auxiliar semana 2

P1) En el modulo triangulo.py se encuentran las siguientes funciones:

- **esTriangulo**: Entrega un booleano verificando si 3 lados pueden formar un triángulo
- **perimetroTriangulo**: Entrega el valor del perímetro de un triángulo dado sus 3 lados
- **areaTriangulo**: Entrega el valor del área del triángulo dados sus 3 lados

Programa la función **datosTriangulo** usando el modulo triangulo.py donde dado 3 valores verifica si es un triangulo, si es así muestra en pantalla el valor del perímetro y área, en caso contrario imprime en pantalla, no es un triángulo. Siga el siguiente esquema:

```
datosTriangulo(2,3,4)
perimetro: 9
area: 2.9047375096555625
```

```
datosTriangulo(8,1,2)
no es un triangulo
```

```
# codigo
```

```
# Esto tiene formato de código
```

P2) Queremos realizar un programa que verifique si un numero a es **diminutivo** de otro numero b otro, es decir , si cada uno de los dígitos de a es menor estricto a cada uno de los dígitos de b

Ejemplo: 523 es diminutivo de 687. En cambio 418 no es diminutivo a 293

Cree la función diminutivo(a,b), en donde a y b son números enteros, cada uno de 3 cifras. Para esto siga el siguiente comportamiento:

- Si a es diminutivo de b, entonces la función debe mostrar "correcto" en pantalla retornando un numero 1.
- En caso contrario, debe mostrar "incorrecto! n_a es mayor o igual a n_b", donde n_a y n_b son los dígitos donde no se cumple la condición y finalmente retornar un numero 0

```
# codigo
```

P3) En plena cuarentena te encuentras en tu casa a punto de entregar una tarea para el ramo de programación, pero ha ocurrido un problema, te has quedado sin internet, a lo que tienes que elegir una de las siguientes alternativas:

- Correr donde tu amigo más cercano (amigo_x,amigo_y) para pedirle internet y enviar tu tarea
- Ir donde tu abuela (abuela_x,abuela_y) para conseguir internet y lograr enviar tu tarea

Para esto necesitas calcular el lugar más cercano y así tomar una buena decisión. Además cuentas con un permiso de 30 minutos (si no iras detenido), consideramos que la velocidad sera de 15km/h en bicicleta (trabaje en minutos, las distancias están en km). Para lograrlo siga los siguientes pasos:

A)

- i) **distancia(x1,x2,y1,y2)**: Esta calcula la distancia entre dos puntos
- ii) **tiempo(distancia)**: Esta debe calcular el tiempo que demora en recorrer esa distancia

```
# codigo
```

B) Realice un programa interactivo que solicite su ubicación, la ubicación de su amigo y de su abuela, finalmente entregue como resultado la ubicación más cercana y si es posible llegar a tiempo sin que el permiso expire. Para esto siga la siguiente plantilla:

```
Su ubicacion x? 5
Su ubicacion y? 6
Amigo x? 5
Amigo y? 5
Abuela x? 7
Abuela y? 9
Menor distancia: amigo
Tiempo:3.75
Lo logra
```

```
# codigo
```

P4) Para esta pregunta vamos a recrear la caracola mágica de Bob Esponja pero con algunas modificaciones. Nuestra caracola tendrá tres respuestas para nuestras preguntas, estas son "Si!",

"No!" o "No he entendido la pregunta". Para esto, programe una función caracola(pregunta) la cual entregara la respuesta de la siguiente manera:

- El 45% de las veces responderá con un "Si!"
- El 35% de las veces responderá con un "No!"
- El resto del 20% responderá con un "No lo he entendido"

Además de esta función, realice un programa interactivo para poner a prueba la caracola, siga el siguiente ejemplo:

```
Cual es la pregunta: ¿Pasare este ramo?  
Respuesta: Si!
```

```
# codigo
```