

Auxiliar 4: Todo lo visto hasta ahora D:

Todos los problemas deben ser resueltos en *Python*, utilizando estrictamente la Receta de Diseño entregada a lo largo del curso. Use nombres apropiados para funciones y variables, y testee cada vez que sea posible.

1. Sismos

1.1. Parte a

Siguiendo la receta de diseño, escriba una función recursiva que reciba un número entero positivo que representa a varios números de dos dígitos (por ejemplo el número 456327 representa los números 45, 63 y 27). La función debe entregar el mayor de los números. Por ejemplo, `mayor(456327)` entrega 63.

1.2. Parte b

El servicio sismológico ha registrado muchos temblores desde el día 16 de septiembre. Al respecto, escriba la función recursiva de encabezamiento `def temblores(diaInicial,diaFinal)`: que lea (con `input`) para cada uno de los días de septiembre (entre `diaInicial` y `diaFinal`) las intensidades de los temblores y escriba (con `print`) el temblor de mayor intensidad del día (usando la función `mayor`). La función `temblores` no retorna ningún valor y debe establecer un diálogo como el indicado en el siguiente ejemplo para la invocación `temblores(16,24)`:

```
dia 16: intensidades de los temblores? 50847161687656
mayor fue 84
dia 17: intensidades de los temblores? 5458576657575253515350505260575353555251
mayor fue 66
...
dia 24: intensidades de los temblores? 48464448455448
mayor fue 54
```

Nota: Las intensidades de los temblores de cada día se ingresan en un solo número entero. Por ejemplo el número 50847161687656 representa temblores de intensidades 5.0, 8.4, 7.1, 6.1, 6.8, 7.6 y 5.6.

2. Series

Uno de los resultados más bacanes y útiles del cálculo es la Serie de Taylor, la cual permite representar una función como una suma infinita de valores. En particular, en este caso nos interesa la expansión de Taylor para la función *seno*, la cual es igual a :

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

Como esto es una suma infinita, y no tenemos infinito tiempo para calcularla, la aproximaremos hasta cierto umbral. Para ello:

- Primero defina las funciones `cerca(x,y,epsilon)` y `factorial(n)`
- Defina una función llamada `taylor_kesimo(x, k)`, que retorne el k -ésimo valor de la serie de Taylor de $\sin(x)$
- Defina una función recursiva `seno_epsilon(x, k, valor_kesimo, epsilon)`, que calcule la serie de Taylor de $\sin(x)$ hasta $k + 1$, y se detenga cuando `abs(taylor_kesimo(x, k + 1) - taylor_kesimo(x, k)) <= epsilon`
- Finalmente, programe una función `seno(x, epsilon)`, que retorne $\sin(x)$.