

Auxiliar 4: Problemas de aproximación y datos compuestos

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.

Frases clave:

- `cerca(x,y,epsilon)`
- Estructura de datos inmutable
- `Estructura.py`
- Datos compuestos

1. Numero de euler

La constante matemática e , es uno de los números irracionales más importantes. Es aproximadamente igual a 2,71828 y aparece en diversas ramas de la matemática, al ser la base de los logaritmos naturales y formar parte de las ecuaciones del interés compuesto y otros muchos problemas.

Una de las definiciones de la función exponencial es la serie:

$$e^x = 1 + \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Usando la serie de la función exponencial dada, implemente una función que calcule el número e recursivamente, hasta que la diferencia de la aproximación de dos llamadas recursivas sea menor a `epsilon`.

2. El tiempo

Escriba un módulo que implemente la estructura `Tiempo`: `horas(int)` `minutos(int)` con las siguientes funciones:

- `crearTiempo`, que dada una cantidad de horas y minutos, crea un "Tiempo correcto". Por ejemplo, `crearTiempo(0,75)` retorna un `Tiempo` de 1 hora y 15 minutos, **NO** un `Tiempo` de 0 horas y 75 minutos.
- `aString` que dado un `Tiempo` entrega la hora que representa en formato `HH:MM`
- `enMinutos` que dado un `Tiempo` entrega un `int` que equivale al tiempo expresado únicamente en minutos.
- `comparar` que recibe dos tiempos y retorna 1 si el primero es mayor que el segundo, 0 si son iguales y -1 si el segundo es mayor. Considere que a las 23 horas es más tarde que a las 0 horas.
- `diferencia` que recibe dos `Tiempo` y retorna un `Tiempo` que representa la diferencia mínima entre las horas que representan. Por ejemplo, la diferencia entre un `Tiempo` con 17 horas y 45 minutos con uno con 16 horas y 30 minutos, debe ser de 1 hora y 15 minutos.

3. Feria de la introducción a la programación

La famosísima feria, programapalooza, busca que los alumnos de la Universidad de Chile se acerquen a la programación. En el lanzamiento de la actividad, decenas de alumnos fueron invitados a jugar y a aprender a programar de una forma didáctica y entretenida. El comite encargado de la inscripción le pide que diseñe un programa para hacer el proceso de inscripción más rápido. En ese punto, se le encomienda que usted que:

- Cree estructuras que permitan guardar datos sobre cada participante. Debe incluir nombre, sexo (H para los niños, M para las niñas) y el año de entrada a la universidad.
 - Como hay demasiados interesados en la feria, la organizacion decidio que solo los alumnos que han entrado en los ultimos 3 años a la universidad pueden participar. Cree una funcion que verifique si un alumno cumple esa condicion o no dado que año es.
 - Cree un programa que pida datos sobre los participantes e indique si cumple los requerimientos, si se finaliza la inscripcion, el programa debe mostrar la cantidad de niños y niñas inscritos en la feria.
- Indicacion: Es muy buena idea hacer una estructura con tres campos para numeros enteros.

Ejemplo del dialogo con el usuario:

```
Año actual? 2017
Nombre? Martin
Sexo? H
Año de ingreso a la UCH? 2016
Bienvenido Martin!
Nombre? Sofia
Sexo? M
Año de ingreso a la UCH? 2014
Bienvenida Sofia!
```

```
Nombre? Carlos
Sexo? H
Año de ingreso a la UCH? 2011
Lo sentimos Carlos, no puedes participar.
Nombre? FIN
Hay 2 participantes inscritos este año
2017!
Niñas: 1, Niños: 1
```