

## Auxiliar 3

23 de septiembre de 2020

Profesores: Nelson Baloian

Auxiliares: Valentina "Gato Negro" Aravena - Pablo "Pingüino" Gutiérrez

Nicolás "Delfín" Rojas - Lucas "Águila" Oyarzún

### Objetivos

- Aprender sobre el uso de módulos
- Entender e implementar funciones recursivas.

Todos los ejercicios deben ser resueltos en *Python*, utilizando la receta de diseño vista a lo largo del curso (salvo que se les indique lo contrario o no sea aplicable). Use nombres apropiados para sus funciones y variables, y testee cada vez que sea posible.

- P1.** Crear una función recursiva, llamada  $mcd\_conMin(x,y,min)$ , que retorna el máximo común divisor entre  $x$  e  $y$ , recibiendo como tercer parámetro el mínimo entre estos dos números.

Ahora utilizando el Algoritmo de Euclides que se muestra a continuación, cree la función  $mcd(x,y)$  para calcular el máximo común divisor entre  $x$  e  $y$ .

$$mcd(a, 0) = a \quad (1)$$

$$mcd(a, b) = mcd(b, a \bmod b) \quad (2)$$

- P2.** Cree un programa interactivo que pida ingresar números hasta que se ingrese un número negativo, ahí tiene que entregar el mayor número entre todos los números ingresados

- P3.** Los números *N-drónicos* son números como cualquier otro, pero poderosos! ya que cuentan con la siguiente propiedad: son múltiplos de la suma de sus dígitos, pero solo si la suma de sus dígitos es mayor a 1. El Sr. Luckys le pide crear un programa, que le entregue un número e indique si es N-drónico o no. Para esto:

- a) Cree una función recursiva llamada **sumarDigitos(n)**, que recibe un numero entero positivo, y entrega como resultado la suma de los dígitos de tal numero.
- b) Cree una función llamada **Ndrónico(n)**, que recibe un numero entero, y entrega como resultado un valor booleano que indica si el numero es N-drónico o no.

- P4.** Implemente el juego de cachipun, para ello cree las siguientes funciones:

- $generarMovimiento()$ : Que genera un movimiento aleatorio, que puede ser "piedra", "papel" o "tijeras". (No incluir test)
- $verGanador(jugador1, jugador2)$ : Que recibe los movimientos de dos jugadores (en formato string) y entrega cómo resultado el número 1 si ganó el jugador1, 0 si hubo un empate o -1 si ganó el jugador2.

- *jugarNveces(N)*: Función recursiva que realice N juegos de cachipun, en donde en cada iteración le pregunte al jugador cuál jugada va a realizar y la compare con una jugada aleatoria realizada por el Computador, para luego indicar cual fue el ganador de dicho juego. Finalmente entrega cómo puntaje final: juegos ganados - juegos perdidos. (No incluir test)

Luego cree un juego interactivo que utilice el modulo anterior en donde se pregunte la cantidad de juegos que se quieren realizar, e indicar si al final hubo un empate, ganó o perdió contando todos los juegos realizados.

```
¿Cuántas veces quieres jugar? 3
¿Que jugada va a realizar?(piedra,papel,tijera): piedra
Jugaste: piedra
El computador jugó: tijera
Ganó el jugador
¿Que jugada va a realizar?(piedra,papel,tijera): papel
Jugaste: papel
El computador jugó: papel
Hubo un empate
¿Que jugada va a realizar?(piedra,papel,tijera): tijera
Jugaste: tijera
El computador jugó: papel
Ganó el jugador
Resultado final: ¡¡GANASTE!!
```