Abstracción de datos

Sección cátedra:		Grupo N°:	
Integrantes **presentes**:			

Pregunta 1

Con tu grupo de trabajo, elijan una de las representaciones mostradas en la figuras 1 e implementen en Python la función `hayJugadaGanadora(tablero)`.

Pregunta 2

Crea una lista llamada `listaPrueba` con los valores 4, 6, 1, -8, 9, 0 y -2 e indica cual es el resultado de las siguientes expresiones y por qué.

a) `calculaMinimoYMaximo(listaPrueba)`
b) `valor1, valor2 = calculaMinimoYMaximo(listaPrueba)`
c) `valor1, valor2, valor3 = calculaMinimoYMaximo(listaPrueba)`
d) `valor1 = calculaMinimoYMaximo(listaPrueba)[0]`
e) `valor1 = calculaMinimoYMaximo(listaPrueba)`

Pregunta 3

Para una aplicación importante se necesita contar con el tipo de datos “número racional”. Este tipo de dato ha de proveer las funciones y cumplir las propiedades especificadas en la siguiente tabla. Implemente el tipo de dato `numeroRacional` usando tuplas de Python.

Ejercicio propuesto

Agrega la función `simplicaR(r)` que simplifique el número racional r y que se encargue de manejar los signos (por ejemplo: -7/-4 ⇒ 7/4, 8/-6 ⇒ -8/6). Usa esta función antes de cada sentencia de retorno de funciones implementadas en el problema 3.

`creaR(a, b)`	Entrada: números enteros a y b Salida: el `numeroRacional` a/b
`sumaR(r1, r2)`	Entrada: dos valores `numeroRacional` r1 y r2 Salida: el `numeroRacional` r1+r2
`restaR(r1, r2)`	Entrada: dos valores `numeroRacional` r1 y r2 Salida: el `numeroRacional` r1-r2
`multiplicaR(r1, r2)`	Entrada: dos valores `numeroRacional` r1 y r2 Salida: el `numeroRacional` r1·r2
`ponderaR(i, r)`	Entrada: un valor valor entero `i` y un valor `numeroRacional` r Salida: el `numeroRacional` que resulta de multiplicar el número racional r por el entero `i`.
`stringR(r)`	Entrada: un valor `numeroRacional` r Salida: un string con la representación de r en un formato conveniente para el usuario. Por ejemplo: `stringR(4/-7)` ⇒ '-4/7' `stringR(14/7)` ⇒ '2' `stringR(-16/7)` ⇒ '-2 2/7'
Propiedades	Sean a, b, c, d y e números enteros, c y d distintos de cero: • r1=a/b y r2=c/d son números racionales • Si r=a/b ⇒ e·r=(e·a)/b • Si r1=a/b y r2=c/d ⇒ r1+r2 = (a·e+c·f)/mcd(r1, r2) donde mcd(r1, r2) es el mínimo común denominador de r1 y r2, y e=mcd(r1, r2)/b y f=mcd(r1, r2)/d son números enteros • Si r1=a/b y r2=c/d ⇒ r1·r2 = a·c/b·d

Ayuda

```
# Calcula el máximo común divisor (MCD) de dos valores enteros
# Entrada: enteros a y b
# Salida: MCD(a, b)
def MCD(a, b):
   assert type(a) == int, "primer valor no es un entero: %r" % a
   assert type(b) == int, "segundo valor no es un entero: %r" % b
   if b == 0:
      return a
   else:
      return MCD(b, a % b)

# Calcula el mínimo común múltiplo (mcm) de dos valores enteros
# Entrada: enteros a y b
# Salida: mcm(a, b)
def mcm(a, b):
   assert type(a) == int, "primer valor no es un entero: %r" % a
   assert type(b) == int, "segundo valor no es un entero: %r" % b
   return abs(a * b) / MCD(a, b)

# Calcula el mínimo común denominador (mcd) de dos valores racionales
# Entrada: racionales r1 y r2
# Salida: mcd(r1, r2)
def mcd(r1, r2):
   return mcm(r1[1], r2[1])
```