

# LABORATORIO ★ 11

## RECURSIÓN

**Instrucción :**

- *El presente ejercicio tiene una bonificación en décimas para la siguiente evaluación individual del curso. La bonificación estará en el rango 0 - 5 décimas. Se subirá solución del ejercicio propuesto dentro de una semana.*

## Problema

Resolver una ecuación diofántica de la forma:

$$a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots + a_n \cdot x_n = C$$

consiste en encontrar valores dentro de un conjunto  $D$  de números enteros para las variables  $x_1, \dots, x_n$ , dados los valores de  $a_1, \dots, a_n$  y el valor de  $C$ . Escribe una función para resolver este tipo de ecuaciones, la que debe ser de tipo recursiva (funciones sólo iterativas no serán evaluadas). De haber una solución, tu función deberá imprimir en pantalla una lista con los valores de  $x_1$  hasta  $x_n$ . En caso contrario, deberá imprimir “no hay solucion”.

Tu función deberá recibir una lista no vacía de números enteros para los coeficientes  $a_1, \dots, a_n$ , y un número entero para la constante  $C$ . Además, deberá suponer que los valores de  $x_i$  son números enteros entre  $-10$  y  $10$ , es decir,  $D = \{-10, -9, \dots, 9, 10\}$ . Por ejemplo, si quieres resolver la siguiente ecuación diofántica

$$2x_1 + 3x_3 = 5$$

deberás llamar a tu función usando como argumentos la lista  $[2, 3]$  y el número  $5$ , y debería imprimir en pantalla, por ejemplo,  $[-8, 7]$ , ya que  $2x(-8) + 3x(7) = 5$ . Nota que puede haber más de una solución; basta con que el programa imprima una solución válida (si la hay), no todas las soluciones.

(**Solución**) Para este ejercicio introduciremos el concepto de *backtracking*, el cual recorre todas las posibles soluciones de un árbol de posibilidades. Para cada  $x$  de la ecuación tenemos 21 posibles valores, los cuales serán evaluados en la función. Primero definimos la función que resuelve la ecuación dado los parámetros pedidos: lista de valores de  $a$ , el valor de  $C$  y una lista de valores de  $x$ , la cual por defecto, es vacía.

---

```
1 def resolver(listaA,C,listaX):
2     if len(listaA) == len(listaX):
3         suma = 0
4         for i in range(0,len(listaA)):
5             suma += listaA[i]*listaX[i]
6         if suma == C:
7             return True
8         return False
9     else:
10        for i in range(-10,10+1):
11            listaX.append(i) # Se pone valor a 'x'
12            if resolver(listaA,C,listaX):
13                return True
14            listaX.pop()
15        return False
```

---

Finalmente definimos una función que imprima un **único** resultado, como pide el enunciado, en caso de pedir todas las soluciones, la función que cambia es esta y un par de líneas en resolver

---

```
1 def diofantica(listaA,C):
2     listaX = []
3     if resolver(listaA,C,listaX):
4         print(listaX)
5     else:
6         print("No tiene solución")
```

---