Examen de Algorítmica y Complejidad (Plan 2014) — **22 de diciembre de 2022**

Nº matrícula:_____ **Nombre:** _____

Apellidos: _____

1) **Problema** (**2.5 puntos**). La empresa Olive.S.A produce y distribuye aceite de oliva a domicilio. Con el tiempo la empresa se ha hecho con una larga lista de clientes y necesitan un nuevo programa informático para decidir a qué clientes vender y distribuir con el objetivo de **maximizar sus beneficios**. A diario la empresa dispone de un depósito de T litros de aceite para vender y distribuir, y de una lista de clientes que incluye la siguiente información: los litros de aceite que solicita (*litros*), el precio al que paga el litro (*precioLitro*) y el gasto que le supone a la empresa distribuir el producto al cliente (*gasto*), entre otras razones debido al precio de la gasolina y la distancia que debe recorrer para la entrega.

   Ejemplo: Consideremos que disponemos de T=155 litros de aceite para vender y los siguientes datos de los clientes:

CLIENTE	**A**	**B**	**C**	**D**	**E**	
litros	30	128	110	5	22	*litros*
precioLitro	5	9	10	3	5	€
gasto	6	50	5	12	15	€

   La solución sería un beneficio de 1242€, que se obtienen al distribuir a los clientes A, C y D los litros solicitados:
   ((30*5 - 6)+(110*10 - 5)+(5*3 - 12)) = (144+1095+3)=1242

   Para resolver este problema, se pide implementar un algoritmo en Java basado en Programación dinámica con la siguiente cabecera:

```
int maximoBeneficio(int[] litros, int[] precioLitro, int[] gasto, int T)
```

   a) (**0.25 puntos**) Define la **entrada**, la **salida** y la **semántica** de la función sobre la que estará basado el algoritmo de programación dinámica.

MaxBenef(i,x): beneficio máximo que puede conseguir la empresa cuando tiene pedidos de los clientes 0..i y dispone de x litros de aceite para vender (i=0 indica que sólo se dispone del cliente 0, j=0 indica que no se dispone de aceite para vender).

   b) (**0.5 punto**) Expresa recursivamente la función anterior.

Si x>=0 (i=0 -> sólo se dispone del cliente 0)
  MaxBenef(0, x) = 0, si x<litros[0]
  MaxBenef(0, x) = litros[0]*precioLitro[0]-gasto[0]), si x>=litros[0]

Si i>=0 (x=0 -> no se dispone de aceite para vender)
  MaxBenef(i, 0) = 0

Si i>0 y 0<x<litros[i]
  MaxBenef(i, x) = MaxBenef(i-1, x)

Si i>0 y x>=litros[i]

MaxBenef(i, x) = max{ MaxBenef(i-1, x) ,
(litros[i]*precioLitro[i]-gasto[i]) + maxBenef(i-1, x-litros[i]}

c) (**1.75 puntos**) Basándote en los anteriores apartados implementa el algoritmo en Java siguiendo el esquema de *Programación Dinámica*.

```
private int maximoBeneficio(int[] litros, int[] precioLitro, int[] gasto, int T){

    int[][] maxBenef = new int[litros.length][T+1];
    for (int x=0; x<=T; x++)
        if (litros[0]>x) maxBenef[0][x]=0;
        else maxBenef[0][x] = (litros[0]*precioLitro[0])-gasto[0];
    for (int i=0; i<litros.length; i++) maxBenef[i][0]=0;

    for (int i=1; i<litros.length; i++){
        for (int x=1; x<=T; x++){
            if (litros[i]>x)
              maxBenef[i][x] = maxBenef[i-1][x];
            else
              maxBenef[i][x] = Math.max(maxBenef[i-1][x],
                                        ((litros[i]*precioLitro[i])-gasto[i])+
                                         maxBenef[i-1][x-litros[i]]);
        }
    }
    return maxBenef[litros.length-1][T];
}
```