Hoja de ejercicios sobre subprogramas recursivos
Programación I / II

Escuela Universitaria de Estadística
Depto. Sistemas Informáticos y Computación
Universidad Complutense de Madrid

Define subprogramas recursivos para los siguientes cálculos. Incluye cada subprograma en un programa que efectúe pruebas de funcionamiento adecuadas. (En los primeros se ha escrito un paréntesis o una pista para facilitar el planteamiento del caso o casos recurrentes.)

1. sumaAritmetica(n) $= (1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n-1} + \frac{1}{n}) = (1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n-1}) + \frac{1}{n}$
2. potencia(x, n) $= (x^n) = x(x^{n-1})$
3. Otra versión de la función potencia:

$$\begin{array}{lcll} x^0 & = & 1 & \\ x^n & = & (x*x)^{n/2} & \text{, si } n > 0 \text{ y es par} \\ x^n & = & x*x^{n-1} & \text{, si } n > 0 \text{ y es impar} \end{array}$$

4. La *raíz digital* de un número natural, o sea, la suma de sus cifras: el propio número n si es de una cifra; la cifra de las unidades + la suma de las cifras de `n div 10` en caso de que el número tenga más de una cifra.
5. La cifrá k-ésima de un número natural, siendo la primera la de las unidades, las decenas la segunda, etc.
6. La función máximo común divisor de dos números naturales, descrito así:
   - $mcd(a, a) = a$
   - $mcd(a, b) = \left\{ \begin{array}{ll} mcd(a, b-a) & \text{si } b > a \\ mcd(a-b, b) & \text{si } b < a \end{array} \right\}$
7. El coeficiente binomial, definido así:

$$\begin{array}{ll} \begin{pmatrix} m \\ 0 \end{pmatrix} & = \begin{pmatrix} m \\ m \end{pmatrix} = 1 \\ \begin{pmatrix} m \\ n \end{pmatrix} & = \begin{pmatrix} m-1 \\ n \end{pmatrix} + \begin{pmatrix} m-1 \\ n-1 \end{pmatrix} \end{array}$$

   Calcula a mano, paso a paso $\begin{pmatrix} 5 \\ 2 \end{pmatrix}$. Forma la pila recursiva y recoge dos o tres estados. Forma también el árbol de llamadas recursivas. Explica cuál de estas dos estructuras está relacionada con el espacio de memoria y cuál con el tiempo de ejecución.
8. Usa los hechos siguientes

$$\begin{array}{ll} \sum_{i=a}^{a} x_a & = x_a \\ \sum_{i=a}^{b} x_i & = \sum_{i=a}^{m} x_i + \sum_{i=m+1}^{b} x_i \end{array}$$

   para definir recursivamente el cálculo de $\sum_{i=a}^{b} \frac{1}{x^{2i-1}}$.
9. Cuando a y b están muy próximos (no más lejos que una centésima, digamos),

$$\int_a^b f(x) \simeq (b-a)f(m)$$

   Y cuando están más alejados, podemos usar el hecho siguiente,

$$\int_a^b f(x) = \int_a^m f(x) + \int_m^b f(x)$$

   siendo $m = \frac{a+b}{2}$ para conseguir dos integrales con los límites más cercanos. Usa estos hechos para definir recursivamente el cálculo de $\int_0^\pi sen(x)$.
10. Sabemos que "cero es par" es una afirmación cierta, y también que "cero es impar" es falsa.

```
esPar(0)    = true
esImpar(0)  = false
```

   La paridad cualquier otro número $n > 0$ es la contraria que la del número anterior, $n-1$:

```
esPar(n)    = esImpar(n-1)
esImpar(n)  = esPar(n-1)
```

   (a) Usa estos hechos para definir sendas funciones recursivas que comprueban la paridad de cualquier número natural. (b) Se podría replantear esto con `esPar(n) = esPar(n-2)` y lo mismo para impar?
11. **Ojo: recursividad con vectores** Calculemos recursivamente el máximo elemento de un vector $\overline{v}$ y su posición entre dos posiciones dadas, a y b, así: Si $a = b$, la posición es a y el valor v_a. Si $a < b$, consideramos m (el punto medio entero ente a y b) y entonces procedemos así:
    - Calculamos el máximo elemento y posición de $\overline{v}$ entre a y m: `(maxIzda, posMaxIzda)`
    - Calculamos el máximo elemento y posición de $\overline{v}$ entre $m+1$ y b: `(maxDcha, posMaxDcha)`

    y luego, según sea mayor `maxIzda` o `posMaxDcha` contestamos un par o el otro.