

Trabajo Práctico Nº 06

Tema: Recursividad

1. Dado el siguiente método:

```
static int puzle (int base, int limite)
{
    if (base > limite) return -1;
        else if (base == limite)
            return 1;
        else return base * puzle(base+1,limite);
}
```

1.1 Identificar:

- el caso(s) base del método puzle.
- el caso(s) general del método puzle.

1.2 Mostrar que escribiría con las siguientes llamadas al método recursivo:

- System.out.println(puzle(14,10));
- System.out.println(puzle(4,7));
- System.out.println(puzle(0,0));

2. Mostrar lo que retornará el siguiente método recursivo después de las llamadas dadas:

```
static int concurso(int base, int lim)
{
    if (base == lim)
        return 1;
    else if (base > lim)
        return 0;
    return base + concurso(base+1,lim);
}
```

- x = concurso(0,3) x es
- y = concurso(10,7) y es
- z = concurso(5,50) z es

3. En cada uno de los siguientes métodos identificar:

- el caso(s) base.
- una llamada recursiva a una versión más pequeña del método.
- y explicar que hace cada método.

3.1. **static int** potencia(int m, int n)

```
{
    if (n == 0) return 1;
        else return n * potencia(m,n -1);
}
```

3.2. **static int** fact (int n)

```
{
    if (n > 0) return n * fact(n -1);
        else if (n == 0) return 1;
        else return -1;
}
```

3.3. **static void** clasificar(int [] datos, int desdeind, int hastaind)

```
{
```

Trabajo Práctico Nº 06

```
int maxi;
if (desdeind != hastaind)
{
    maxi = maxposicion(datos,desdeind,hastaind);
    cambiar(datos,maxi,hastaind);
    clasificar(datos,desdeind,hastaind-1);
}
}
```

4. Dado el siguiente método recursivo:

```
static void desconocido(int N, int despl)
{
    int k;
    if (N > 0)
    {
        desconocido (N-1, despl.+1);
        for (k = 1; k <= despl; k++)
            System.out.println(" ");
        for (k = 1; k <= despl; k++)
            System.out.println("1");
        System.out.println(" ");
        desconocido(N-1, despl+1);
    }
}
```

- a) ¿qué hace el método?
- b) ¿Hacer un seguimiento con la llamada desconocido(4,1).

5. Explique por qué el siguiente método puede producir un valor incorrecto cuando se ejecute:

```
static long factorial (long n)
{
    if (n == 0 || n == 1)
        return 1;
    else
        return n * factorial (--n);
}
```

6. ¿Cuál es la secuencia numérica generada por el método recursivo f siguiente?

```
static long f (int n)
{
    if (n == 0 || n == 1)
        return 1;
    else
        return 3 * f (n-2) + 2 f (n - 1);
}
```

7. ¿Cuál es la secuencia numérica generada por el método recursivo f siguiente?

```
static long f (int n)
{
    if (n == 0 || n == 1)
        return 1;
    else
        return 3 * f (n-2) + 2;
}
```

Trabajo Práctico Nº 06

8. ¿Cuál es la secuencia numérica generada por el método recursivo f siguiente?

```
static long f (int n)
{
    if (n == 0 )
        return 1;
    else if (n == 1 )
        return 2;
    else
        return 2 * f (n-2) + f (n - 1);
}
```

9. Escribir un método recursivo que calcule la suma:

$$1 + 2 + 3 + \dots + (n-1) + n$$

10. Escribir un método recursivo para encontrar la suma de los enteros pares hasta N.

$$2 + 4 + 6 + \dots + (n-2) + n$$

11. Escribir un método recursivo para calcular la suma de la siguiente serie de enteros, de 1 hasta N:

$$1^n + 2^{n-1} + 3^{n-2} + \dots + n-1^2 + n^1$$

12. Suponer que la función G está definida recursivamente de la siguiente forma:

$$G(x,y) = \begin{cases} 1 & \text{si } x = y \\ G(x-y+1)+1 & \text{si } y < x \end{cases}$$

Si x, y son enteros positivos.

- escribir el método recursivo
- encontrar el valor de G(8,6)
- encontrar el valor de G(100,10)

13. Sea H(x) un método definido recursivamente $\forall x > 0$, si o x un entero:

$$H(x) = \begin{cases} 1 & \text{si } x = 1 \\ H(x/2)+1 & \text{si } x > 1 \end{cases}$$

- escribir el método recursiva
- encontrar el valor de H(80)
- ¿cómo podemos describir lo que hace este método?

14. Escribir un metodo recursivo *int vocales(String cd)* para calcular el número de vocales de una cadena.

15. Escribir un método recursivo que devuelva un valor true si una palabra pasada como argumento es un palíndromo y devuelve falso en caso contrario.

16. Un ejemplo sobre definición recursiva es la definición de multiplicación de números naturales. El producto de $a * b$ donde a y b son enteros positivos, se puede definir como a sumado así mismo un número de veces igual a b. Esta definición es iterativa. Una definición iterativa equivalente sería:

$$\begin{aligned} a * b &= a & \text{si } b &= 1 \\ a * b &= a * (b-1) + a & \text{si } b &> 1 \end{aligned}$$

Con el fin de evaluar $6 * 3$ utilizando esta definición, debemos primero evaluar $6 * 2$ y entonces al resultado agregarle 6. Para evaluar $6 * 2$, debemos primero evaluar $6 * 1$ y agregarle 6. Pero $6 * 1$ es igual a 6 según la primera parte de la definición. Entonces:



Trabajo Práctico Nº 06

- b) Buscar en el array un determinado valor.
- c) Sumar los elementos del array.
- d) Buscar el valor máximo de los elementos del array.
- e) Buscar el valor máximo en los elementos de un sub-array.

23. Escribir un método recursivo que calcule los números de Fibonacci. La serie se representa de la siguiente manera:

$$\begin{array}{lll} \text{Fib}(n) = & 1 & \text{si } n = 1, n = 2 \\ & \text{Fib}(n-1)+\text{Fib}(n-2) & \text{si } n > 2 \end{array} \quad \begin{array}{l} \text{Fib}(1) = 1 \\ \text{Fib}(2) = 1 \\ \text{Fib}(3) = \text{Fib}(1) + \text{Fib}(2) \\ \text{Fib}(4) = \text{Fib}(2) + \text{Fib}(3) \end{array}$$