

**Ejercicio 1** Implementa una función iterativa (usando un **BUCLE**)  $f$  que, dado un número entero positivo  $n$ , calcule la suma de los  $n$  primeros números enteros positivos. Es decir,  $f(n) = 1 + 2 + 3 + \dots + (n - 1) + n$ .

**Solución:**

```
1 public static int suma_positivos_iterativa(int n)
2 {
3     int s=0;
4     for(int i=1; i<=n; i++)
5         s = s+i;
6
7     return s;
8 }
```

**Ejercicio 2** Implementa una función **RECURSIVA**  $f$  que, dado un número entero positivo  $n$ , calcule la suma de los  $n$  primeros números enteros positivos. Es decir,  $f(n) = 1 + 2 + 3 + \dots + (n - 1) + n$ .

**Solución:**

```
1 public static int suma_positivos(int n)
2 {
3     if (n==1)
4         return 1;
5     else
6         return n + suma_positivos(n-1);
7 }
```

**Ejercicio 3** Implementa una función **RECURSIVA**  $f$  que, dado un número entero no negativo  $n$ , devuelva el número de dígitos de  $n$ . Por ejemplo,  $f(0) = 1$ ,  $f(6) = 1$ ,  $f(95) = 2$ ,  $f(2443) = 4$ .

**Solución:**

```
1 public static int numero_digitos(int n)
2 {
3     if (n<10)
4         return 1;
5     else
6         return 1 + numero_digitos(n/10);
7 }
```

**Ejercicio 4** Implementa una función **RECURSIVA**  $f$  que, dado un número entero no negativo  $n$ , calcule el número entero asociado a la representación de  $n$  en base 2. Por ejemplo,  $f(9) = 1001$  (mil uno). En ese caso,  $9 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 0 + 1$ .

**Solución:**

```
1  public static int decimal_a_binario(int n)
2  {
3      if (n<2)
4          return n;
5      else
6          return 10*decimal_a_binario(n/2) + (n%2);
7  }
```