

Práctica de Recursividad - Soluciones

1) Considera la una función que calcula la suma de los primeros n números impares (positivos):

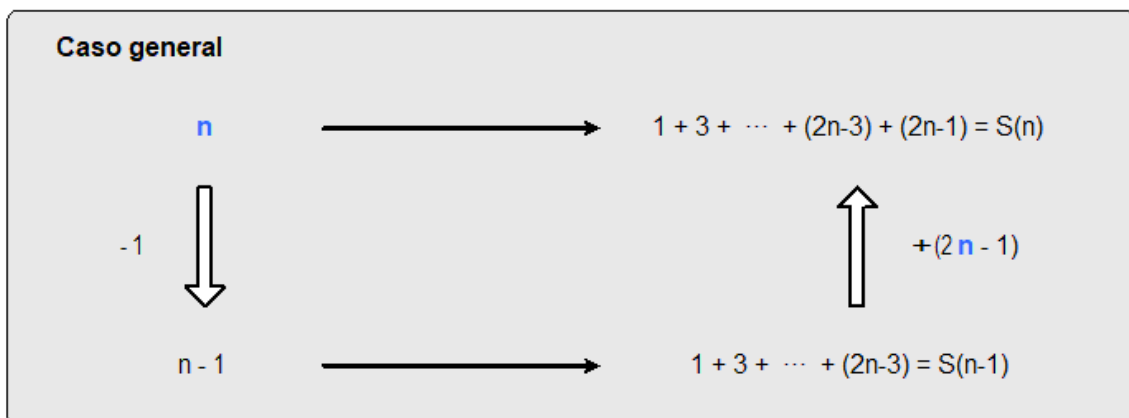
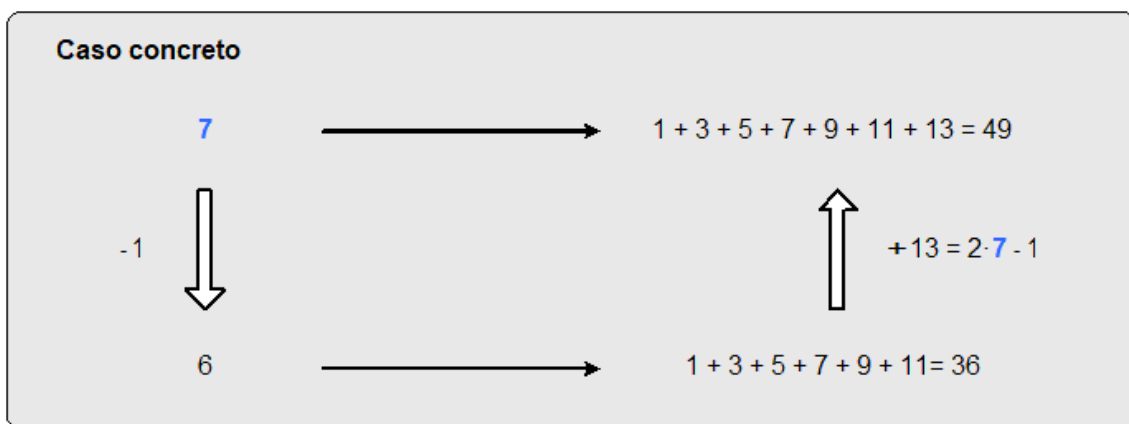
$$S(n) = 1 + 3 + 5 + \dots + (2n-1)$$

Especifica su caso base, y dibuja diagramas para diseñar la función.

Solución:

El caso base es: $S(1) = 1$.

Posibles diagramas:

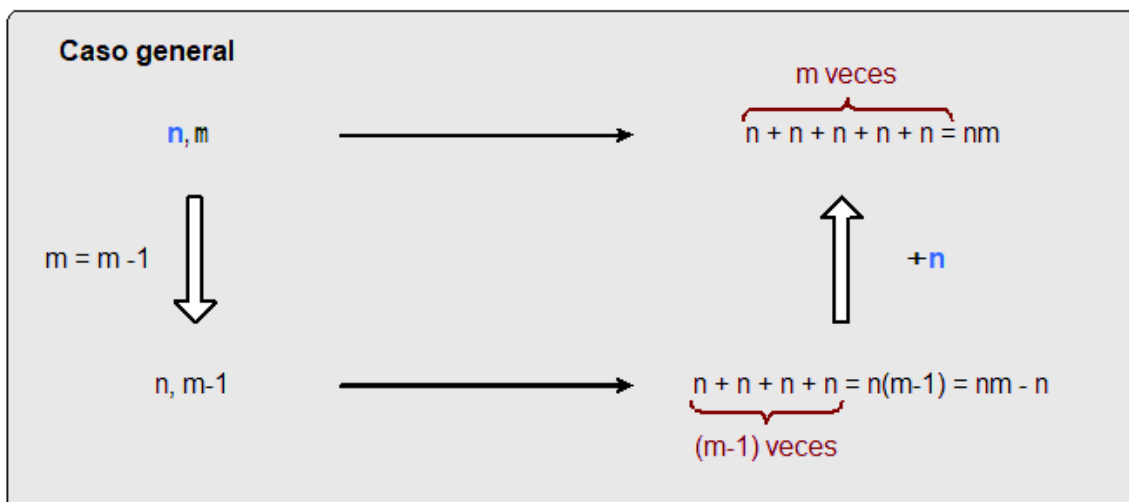
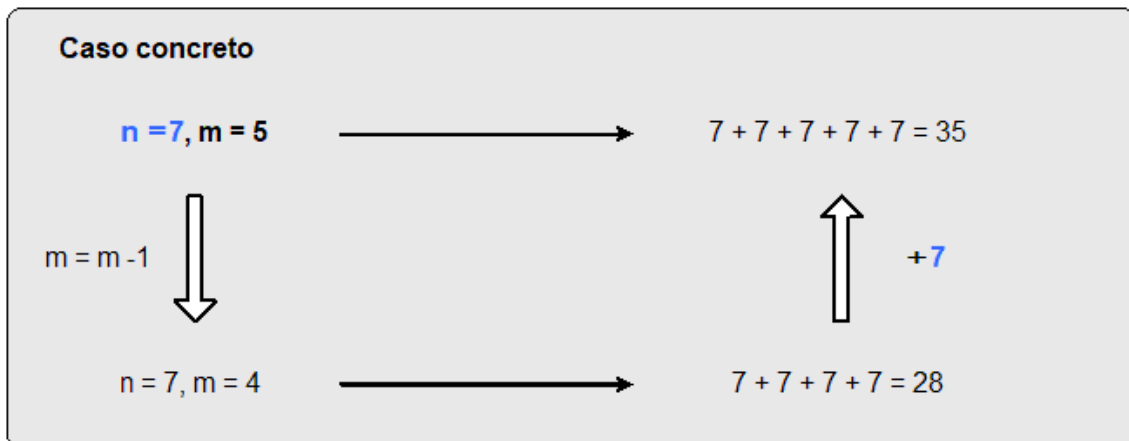


2) Considera una función que calcule, mediante sumas repetidas, el producto de dos números enteros n y m no negativos. La función se podría llamar “producto lento”. Especifica su(s) caso(s) base, y dibuja diagramas para diseñar la función.

Solución:

El caso más sencillo ocurre cuando alguno de los parámetros de entrada es igual a 0. En ese caso la función debe retornar 0. Así que el caso base es $f(n, m) = 0$, si $n=0$ o $n=1$.

Posibles diagramas:

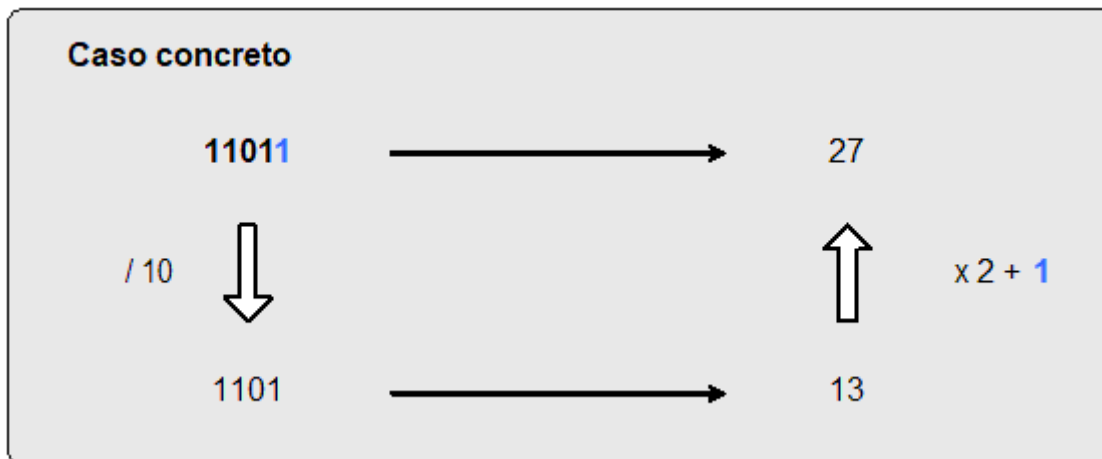
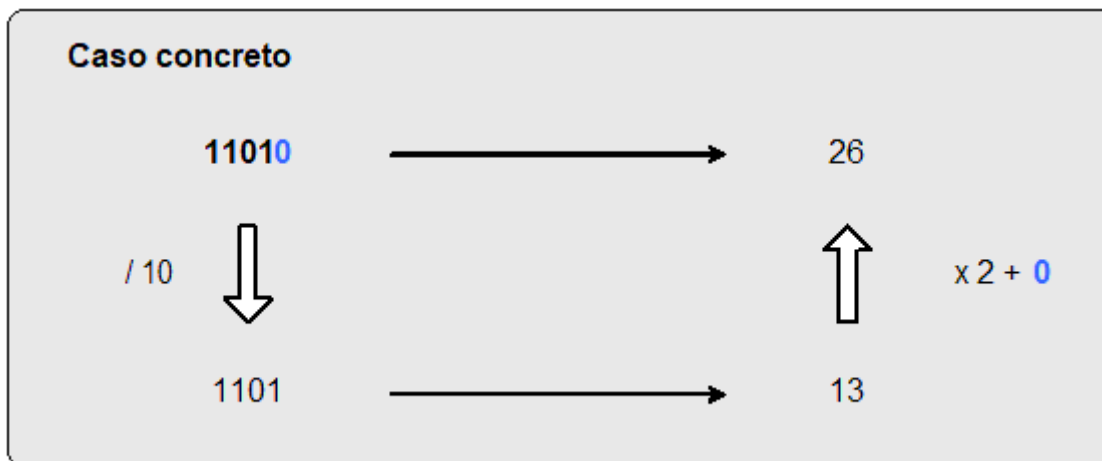


3) Considera una función f que, dado un número entero $n \geq 0$, cuyos dígitos son 0 o 1 necesariamente, calcule el valor del número binario asociado a la secuencia de ceros y unos en n . Es decir, si $n = 11010$, entonces $f(n) = f(11010) = 16 + 8 + 2 = 26$. Especifica su(s) caso(s) base, dibuja un diagrama para diseñar la función, halla el/los caso(s) recursivo(s), implementa la función, y comprueba que funciona con varios casos de prueba.

Solución:

El caso base es $f(n) = n$, si $n < 2$ (si $n=0$ o $n=1$, es decir, si n se puede representar mediante un solo bit).

Posibles diagramas:



Código:

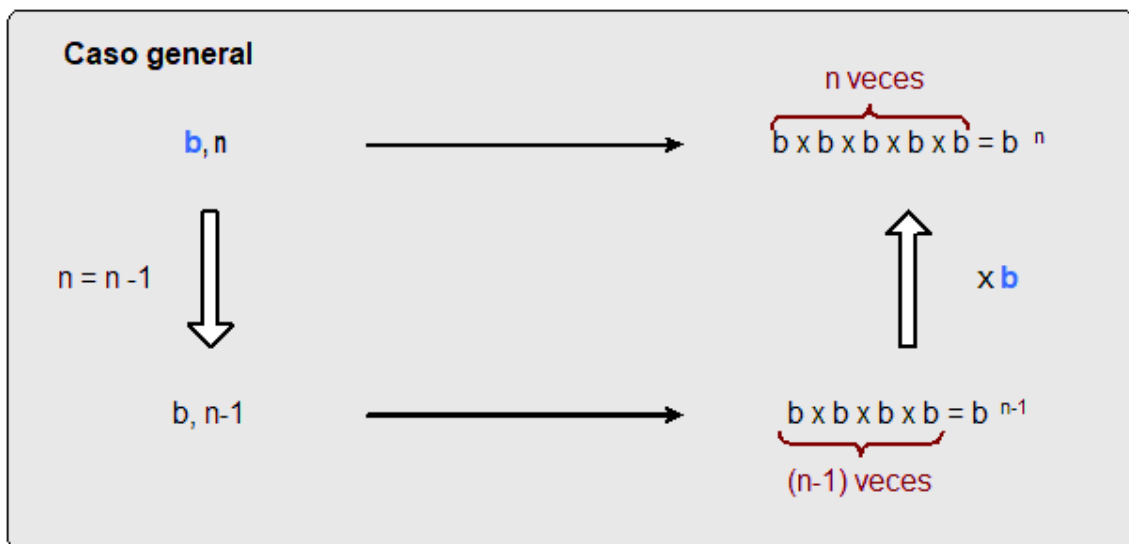
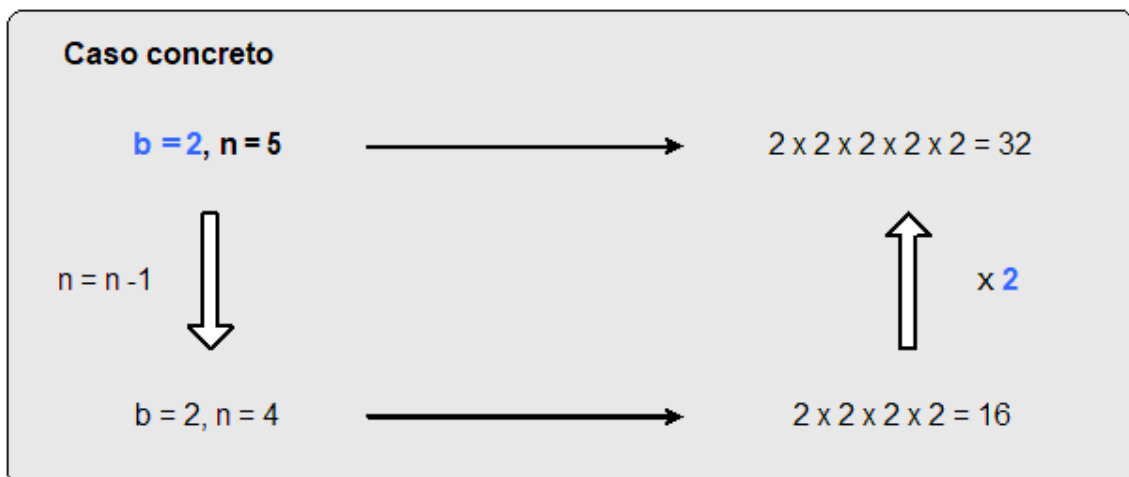
```
public static int binario_a_decimal(int n)
{
    if (n < 2)
        return n;
    else
        return 2 * binario_a_decimal(n/10) + (n%10);
}
```

4) Considera una función que calcule la potencia b^n , donde b es un número real, y n es un número entero no negativo. Especifica su(s) caso(s) base, dibuja un diagrama para diseñar la función, halla el/los caso(s) recursivo(s), implementa la función, y comprueba que funciona con varios casos de prueba.

Solución:

El caso más sencillo ocurre cuando $n = 0$, donde la salida es 1. Por tanto, el caso base es $f(b, n) = 1$, si $n=0$.

Posibles diagramas:



Código:

```

public static double potencia(double b, int n)
{
    if (n==0)
        return 1.0;
    else
        return b * potencia(b, n-1);
}
  
```