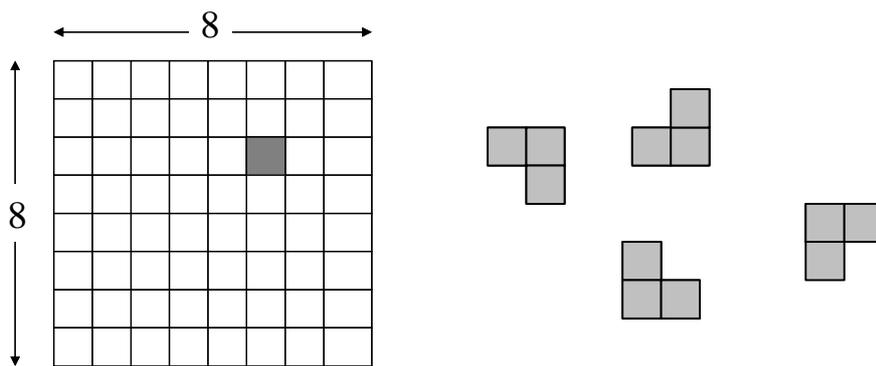


Recursividad

Relación de ejercicios

1. Demuestre por inducción que la función $Q(n)=1^2+2^2+3^2+\dots+n^2$ puede expresarse como $Q(n) = n(n+1)(2n+1)/6$
2. Demuestre por inducción que, para todo n mayor o igual que 1, 133 divide a $11^{n+1} + 12^{2n-1}$
3. Demuestre por inducción que, para todo n mayor o igual que 4, $n! > 2^n$
4. Dado un tablero de ajedrez (de tamaño 8×8) al que le falta una casilla, ¿podemos rellenar las demás casillas utilizando únicamente teselas con forma de L?



PISTA: Realice una demostración por inducción
- ¿Cuántas casillas faltan por rellenar?
- ¿Cuántas teselas se han colocado ya?
- ¿Cuántas casillas quedan tras colocar otra tesela?

Generalice la demostración para un tablero de tamaño $2^n \times 2^n$

5. Demuestre que, con sellos de 4 y 5 céntimos, se puede franquear cualquier carta que requiera sellos por valor de 12 o más céntimos.
6. Un granjero ha comprado una pareja de conejos para criarlos y luego venderlos. Si la pareja de conejos produce una nueva pareja cada mes y la nueva pareja tarda un mes más en ser también productiva, ¿cuántos pares de conejos podrá poner a la venta el granjero al cabo de un año?

FUENTE: *Liber abaci*, sección III

7. Dado el siguiente fragmento de código:

```
static final double N = 2;
static final double PREC = 1e-6;

static double f (double x)
{
    return x*x-N;
}

static double bisect (double min, double max)
{
    double med = (min+max)/2;

    if (max-min<PREC) {
        return med;
    } else if (f(min)*f(med)<0) {
        return bisect (min,med);
    } else {
        return bisect (med,max);
    }
}
```

- a) ¿Qué calcula la llamada a la función recursiva `bisect(0,N)`? Si cambiamos el valor de `N`, ¿qué estaríamos calculando? ¿Y si cambiásemos la función `f(x)`?
- b) Implemente un algoritmo iterativo equivalente.

8. Dado el siguiente algoritmo recursivo:

```
void f(int num, int div)
{
    if (num>1) {
        if ((num%div) == 0) {
            System.out.println(div);
            f(num/div,div);
        } else {
            f(num,div+1);
        }
    }
}
```

- a) Dado un número cualquiera `x`, ¿qué nos muestra por pantalla la llamada a la función recursiva `f(x,2)`? ¿Cuál sería un nombre más adecuado para la función `f`?
- b) Implemente un algoritmo iterativo equivalente.

9. ¿Cuál es el resultado de esta función para distintos valores de x?

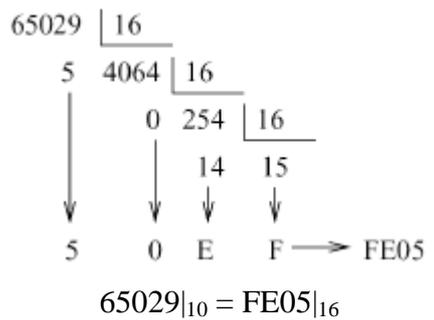
```
static int f (int x)
{
  if (x > 100)
    return (x - 10);
  else
    return (f (f (x+11)));
}
```

10. Construya una función que convierta un número decimal en una cadena que represente el valor del número en hexadecimal (base 16). A continuación, generalice la función para convertir un número decimal en un número en base B (con B<10).

Recordatorio: El cambio de base se realiza mediante divisiones sucesivas por 16 en las cuales los restos determinan los dígitos hexadecimales del número según la siguiente correspondencia:

Resto	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dígito	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Por ejemplo:



11. Implemente, tanto de forma recursiva como de forma iterativa, una función que nos diga si una cadena de caracteres es simétrica (un palíndromo). Por ejemplo, “DABALEARROZALAZORRAELABAD” es un palíndromo.

12. Implemente, tanto de forma recursiva como de forma iterativa, una función que le dé la vuelta a una cadena de caracteres.

NOTA:

Obviamente, si la cadena es un palíndromo, la cadena y su inversa coincidirán.

13. Implemente, tanto de forma recursiva como de forma iterativa, una función que permitan calcular el número de combinaciones de n elementos tomados de m en m. Realice dos versiones de la implementación iterativa, una aplicando la fórmula y otra utilizando una matriz auxiliar (en la que se vaya construyendo el triángulo de Pascal).

14. Implemente, tanto de forma recursiva como de forma iterativa, una función que nos devuelva el máximo común divisor de dos números enteros utilizando el algoritmo de Euclides.

ALGORITMO DE EUCLIDES

Dados dos números enteros positivos m y n , tal que $m > n$, para encontrar su máximo común divisor

(es decir, el mayor entero positivo que divide a ambos):

- Dividir m por n para obtener el resto r ($0 \leq r < n$)
- Si $r = 0$, el MCD es n .
- Si no, el máximo común divisor es $\text{MCD}(n,r)$.

15. La ordenación por mezcla (*mergesort*) es un método de ordenación que se basa en un principio muy simple: se ordenan las dos mitades de un vector y , una vez ordenadas, se mezclan. Escriba un programa que implemente este método de ordenación.

16. Diseñe e implemente un algoritmo que imprima todas las posibles descomposiciones de un número natural como suma de números menores que él (sumas con más de un sumando).

17. Diseñe e implemente un método recursivo que nos permita obtener el determinante de una matriz cuadrada de dimensión n .

18. Diseñe e implemente un programa que juegue al juego de cifras de “**Cifras y Letras**”. El juego consiste en obtener, a partir de 6 números, un número lo más cercano posible a un número de tres cifras realizando operaciones aritméticas con los 6 números.

19. **Problema de las 8 reinas:** Se trata de buscar la forma de colocar 8 reinas en un tablero de ajedrez de forma que ninguna de ellas amenace ni se vea amenazada por otra reina.

Algoritmo:

- Colocar la reina i en la primera casilla válida de la fila i
- Si una reina no puede llegar a colocarse en ninguna casilla, se vuelve atrás y se cambia la posición de la reina $i-1$
- Intentar colocar las reinas restantes en las filas que quedan

20. **Salida de un laberinto:** Se trata de encontrar un camino que nos permita salir de un laberinto definido en una matriz $N \times N$. Para movernos por el laberinto, sólo podemos pasar de una casilla a otra que sea adyacente a la primera y no esté marcada como una casilla prohibida (esto es, las casillas prohibidas determinan las paredes que forman el laberinto).

Algoritmo:

- Se comienza en la casilla $(0,0)$ y se termina en la casilla $(N-1,N-1)$
- Nos movemos a una celda adyacente si esto es posible.
- Cuando llegamos a una situación en la que no podemos realizar ningún movimiento que nos lleve a una celda que no hayamos visitado ya, retrocedemos sobre nuestros pasos y buscamos un camino alternativo.