

Algoritmos y Estructuras de datos

Soluciones a los ejercicios previos al primer examen parcial

- Escribe una implementación *recursiva* de la función `imprime_mitades`, la cual toma un entero no negativo n e imprime la secuencia $n, n/2, n/4, \dots, 1, 0$ (sin las comas). Por ejemplo, para $n=9$ la función debe imprimir 9 4 2 1 0 mientras que para $n=0$ sólo debe imprimir 0. Está prohibido usar ciclos y la biblioteca de C++ excepto las rutinas de escritura (`printf`, `cout`).

```
void imprime_mitades(int n) {
    std::cout << n << " ";
    if (n != 0) {
        imprime_mitades(n / 2);
    }
}
```

- Escribe una implementación *recursiva* de la función `imprime_alternado`, la cual toma un entero no negativo n e imprime los primeros n números de la secuencia 1, 0, 1, 0, 1, ... (sin las comas). Por ejemplo, para $n=3$ la función debe imprimir 1 0 1 mientras que para $n=0$ no debe imprimir nada. Está prohibido usar ciclos y la biblioteca de C++ excepto las rutinas de escritura (`printf`, `cout`).

```
void imprime_alternado(int n) {
    if (n != 0) {
        imprime_alternado(n - 1);
        std::cout << n % 2 << " ";
    }
}
```

- Escribe una implementación *recursiva* de la función `imprime_escalera`, la cual toma dos enteros no negativos k, n (donde $k \leq n$) e imprime la secuencia $k, k+1, \dots, n-1, n, n-1, \dots, k+1, k$ (sin las comas). Por ejemplo, para $k=2, n=5$ la función debe imprimir 2 3 4 5 4 3 2. Está prohibido usar ciclos y la biblioteca de C++ excepto las rutinas de escritura (`printf`, `cout`).

```
void imprime_escalera(int k, int n) {
    std::cout << k << " ";
    if (k < n) {
        imprime_escalera(k + 1, n);
        std::cout << k << " ";
    }
}
```

- Escribe una implementación *recursiva* de la función `logaritmo2`, la cual toma un entero n potencia de 2 y devuelve el logaritmo base 2 de n . Por ejemplo, para $n=32$ la función debe devolver 5. Está prohibido usar ciclos y la biblioteca de C++.

```
int logaritmo2(int n) {
    if (n == 1) {
        return 0;
    } else {
        return 1 + logaritmo2(n / 2);
    }
}
```