

## Lista extra 2

Prezados alunos, segue a lista com cinco exercícios em C, projetados para consolidar o uso exclusivo de laços for, condicionais if (ou operador ternário) e operações de entrada/saída com **printf/scanf**.

Cada problema deve ser implementado com apenas esses recursos; atentem-se às especificações de cada enunciado e submeta seu código até a data informada.

### 1) Máximo Divisor Comum (MDC) por busca decrescente

**Descrição:** Leia dois inteiros positivos, A e B, e encontre o maior divisor comum executando um único laço for que percorre todos os possíveis divisores de forma decrescente, de  $\min(A,B)$  até 1.

**Entrada:** Dois inteiros positivos A B

**Saída:** MDC = X onde X é o maior divisor comum de A e B.

Fonte: Kernighan, B. W. & Ritchie, D. M. The C Programming Language, 2ª ed., Prentice Hall, 1988 — Exercício 2.4.

### 2) Triângulo de Pascal usando cálculo de binomial

**Descrição:** Leia um inteiro N ( $\leq 15$ ) e imprima as primeiras N linhas do Triângulo de Pascal. Para cada elemento  $C(i, j) = i! / (j! * (i-j)!)$ , calcule os fatoriais com laços for.

**Entrada:** Um inteiro N

**Saída:** Cada linha i deve conter os coeficientes binomiais  $C(i,0) C(i,1) \dots C(i,i)$ .

Fonte: Weiss, M. A. Data Structures and Algorithm Analysis in C, 2ª ed., Addison-Wesley, 1997 — Capítulo 2, seção de coeficientes binomiais.

### 3) Números de Armstrong até N

**Descrição:** Um número de Armstrong (**ou narcísico**) é aquele cuja soma de cada dígito elevado ao total de dígitos resulta nele mesmo (por exemplo,  $153 = 1^3 + 5^3 + 3^3$ ). Leia um inteiro N e imprima todos os números de Armstrong de 1 até N. Use for para:

- Iterar i de 1 a N
- Contar dígitos de i (dividindo por 10 em cada iteração)
- Somar cada dígito elevado ao número de dígitos (outro for aninhado para multiplicações sucessivas)

**Entrada:** Um inteiro N

**Saída:** Todos os números de Armstrong  $\leq N$ , separados por espaço.

Fonte: GeeksforGeeks. “Armstrong Number in C” (adaptado). Disponível em: <https://www.geeksforgeeks.org/program-to-find-armstrong-numbers-in-a-range/>.