

# Exercícios de Estruturas de Controle e Repetição III

## 1. Número Perfeito

Escreva um programa em C que determine se um número fornecido pelo usuário é um número perfeito. Um número perfeito é aquele cuja soma de seus divisores próprios (excluindo o próprio número) é igual a ele. Exemplo: 6 é um número perfeito, pois seus divisores próprios (1, 2 e 3) somam 6.

## 2. Somatório de Fatoriais

Escreva um programa em C que calcule a soma dos fatoriais dos números inteiros entre dois valores fornecidos pelo usuário, inclusive. Por exemplo, se o usuário informar 3 e 5, o programa deve calcular  $3! + 4! + 5!$ .

## 3. Primo Gêmeo

Crie um programa em C que encontre o primeiro par de números primos gêmeos (dois números primos cuja diferença é 2) maiores que um número fornecido pelo usuário. Exemplo de primos gêmeos: (11, 13), (17, 19).

## 4. Números de Harshad

Escreva um programa que verifique se um número fornecido pelo usuário é um número de Harshad (ou Niven), ou seja, um número que é divisível pela soma de seus dígitos. Por exemplo, 18 é um número de Harshad porque 18 é divisível por 9 ( $1 + 8$ ).

## 5. Jogo de Adivinhação com Probabilidade Adaptativa

Crie um programa em C que simule um jogo de adivinhação entre o usuário e o computador, onde o computador tenta adivinhar um número que o usuário escolheu. No entanto, o comportamento do computador é adaptativo, ou seja, ele ajusta suas suposições com base no feedback do usuário sobre se o palpite foi muito alto ou muito baixo.

Regras e funcionamento:

1. O usuário escolhe um número secreto entre 1 e 100, sem informar o computador.
2. O computador tenta adivinhar o número fazendo suposições.
3. Para cada suposição, o usuário deve informar se o palpite foi "muito alto", "muito baixo" ou "correto".
4. Com base no feedback, o computador ajusta seu próximo palpite:
  - Se o palpite foi "muito alto", o próximo palpite deve ser menor.
  - Se o palpite foi "muito baixo", o próximo palpite deve ser maior.
  - O programa termina quando o computador adivinhar o número corretamente.
5. O computador deve usar um sistema adaptativo de suposições (sem utilizar arrays, strings, ponteiros ou funções). Uma maneira de fazer isso seria reduzir o intervalo de possíveis valores de acordo com o feedback do usuário.

Exemplo de execução:

- O usuário escolhe o número 42.
- O computador adivinha 50.
- O usuário informa que o palpite é "muito alto".
- O computador então ajusta seu intervalo para 1 a 49 e faz um novo palpite, por exemplo, 25.
- O usuário diz "muito baixo".
- O computador ajusta o intervalo para 26 a 49, e assim por diante até adivinhar 42 corretamente.