**Exercício Final do Nivelamento: Simulador de Jogo de Adivinhação e Calculadora de Estatísticas**

**Enunciado:** Cada grupo deve criar um programa em C que simule um jogo de adivinhação e, ao final, calcule estatísticas dos resultados dos jogadores. O programa deve funcionar da seguinte maneira:

1. O programa deve gerar aleatoriamente um número entre 1 e 100.
2. O jogador (ou grupo de jogadores) terá 10 tentativas para adivinhar o número.
3. A cada tentativa, o programa deve fornecer dicas se o número é maior ou menor que o palpite dado.
4. Se o jogador acertar, o programa deve encerrar as tentativas e informar em quantas jogadas ele conseguiu acertar. Se o jogador não acertar após as 10 tentativas, o jogo termina com uma mensagem informando o fracasso.
5. Após o fim do jogo (acertando ou não), o programa deve pedir ao jogador para informar se deseja jogar novamente ou encerrar.
6. Ao final de todas as partidas, o programa deve calcular e exibir as seguintes estatísticas:
7. Quantas partidas foram jogadas.
8. Quantas partidas foram ganhas.
9. A média de tentativas para acertar o número (nas partidas vencidas).
10. O jogador deve poder escolher uma operação matemática básica (adição, subtração, multiplicação ou divisão) após cada rodada para treinar seus cálculos. O programa deverá usar um switch case para selecionar a operação correta e realizar o cálculo. Exemplo: "Deseja somar, subtrair, multiplicar ou dividir X por Y?"

**Regras:**

* O uso de condicionais if-else e switch case é obrigatório.
* Laços de repetição devem ser usados para as tentativas do jogo e para as opções de jogar novamente.
* O código deve ter uma boa estrutura e estar comentado.
* Crie um fluxograma ou pseudocódigo para organizar sua solução antes de implementar.

Aqui está um exemplo de como vocês podem gerar números aleatórios em C, usando a função rand() e srand() para garantir que os números gerados sejam aleatórios em cada execução do programa:

Texto

Descrição gerada automaticamente

**Explicação:**

1. srand(time(NULL)): Inicializa o gerador de números aleatórios com base no tempo atual, o que garante que os números serão diferentes em cada execução do programa.
2. rand() % 100 + 1: A função rand() gera um número inteiro grande. O operador % 100 faz com que o número fique no intervalo de 0 a 99, e somamos +1 para que o número fique entre 1 e 100.