

**Protocolo de Finalização Nº 000100284109**

O aluno Gultembergue Carlos com RA **25531299-5** finalizou a atividade **ATIVIDADE 3 - ADSIS - LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO - 54\_2025** em **25/11/2025 21:25:39**

**ATIVIDADE 3 - ADSIS - LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO - 54\_2025**

**Período:**10/11/2025 08:00 a 30/11/2025 23:59 (Horário de Brasília)

**Status:**ABERTO

**Nota máxima:**1,00

**Nota obtida:**

**1ª QUESTÃO**

A empresa DataAnalytics está desenvolvendo um sistema de análise de dados que precisa processar grandes volumes de informações numéricas armazenadas em vetores. Uma das funcionalidades críticas é a ordenação de um vetor com 1000 elementos inteiros em ordem crescente. O sistema deve apresentar os dados organizados para facilitar relatórios gerenciais e consultas dos usuários finais. Durante a fase de otimização, a equipe técnica está analisando diferentes algoritmos de ordenação disponíveis. O gerente de projetos solicitou que seja implementado o método de ordenação que, apesar de não ser o mais eficiente em termos de complexidade computacional, é mais fácil de entender e implementar, considerando que a equipe possui programadores de diferentes níveis de experiência.

Com base nos requisitos apresentados e os métodos de ordenação abordados no material, qual algoritmo deve ser utilizado?

**RESPOSTA:** Bubble Sort (método da bolha) por sua simplicidade de implementação.

**2ª QUESTÃO**

A startup EduTech está desenvolvendo um quiz educativo que apresenta múltiplas opções de disciplinas para o usuário escolher. O menu principal oferece 5 opções: 1-Matemática, 2-História, 3-Geografia, 4-Ciências, 5-Sair. Dependendo da escolha do usuário, o programa deve executar uma ação específica para cada disciplina e exibir uma mensagem personalizada. O programador responsável está decidindo entre usar uma sequência de estruturas if-else aninhadas ou uma estrutura mais organizada e legível. A equipe de code review enfatizou a importância de escolher a estrutura que torne o código mais claro e eficiente, especialmente considerando que no futuro podem ser adicionadas mais disciplinas ao menu.

Considerando a situação apresentada e as diretrizes para estruturas condicionais em C, qual estrutura é mais adequada para implementar este menu?

**RESPOSTA:** Estrutura switch-case com break em cada opção.

**3ª QUESTÃO**

Em programação, um **vetor** (ou *array*, em inglês) é uma das estruturas de dados mais fundamentais e simples. Ele é essencialmente uma coleção de variáveis do **mesmo tipo de dado**, armazenadas em posições contíguas (vizinhas) na memória do computador.

Considere o seguinte trecho de código em Linguagem C:

```
#include <stdio.h>

int main() {
    int A[5] = {10, 20, 30, 40, 50};
    int B[5];
    int i;

    for (i = 0; i < 5; i++) {
        if (i % 2 == 0) {
            B[i] = A[i] + 5;
        } else {
            B[i] = A[i] - 5;
        }
    }

    printf("B[2]: %d\n", B[2]);
    printf("B[3]: %d\n", B[3]);

    return 0;
}
```

Imagem 1: Trecho de código em C

Fonte: o autor.

Com base no texto apresentado, analisando o código, qual será a saída exata impressa na tela após a execução deste programa?

RESPOSTA:	B	2
	: 35	3

#### 4ª QUESTÃO

Em C, strings são arrays de caracteres finalizados com o caractere nulo `'\0'`. A linguagem oferece várias funções da bibliotecas para manipulação de strings.

Considerando o texto apresentado, sobre as funções de manipulação de strings em C, avalie as afirmações a seguir.

- I. A função `strlen(s)` retorna o número de caracteres de uma string `s`, excluindo o caractere nulo `'\0'`.
- II. A função `strcat(dest, src)` adiciona o conteúdo da string `src` ao final da string `dest`.
- III. A função `strstr(haystack, needle)` retorna um ponteiro para a primeira ocorrência da string `needle` dentro de `haystack`, ou `NULL` se não encontrar.
- IV. As funções `strupr(s)` e `strlwr(s)` convertem todos os caracteres da string `s` para maiúsculas e minúsculas, respectivamente.

É correto o que se afirma em:

**RESPOSTA:** I, II e III, apenas.

### 5ª QUESTÃO

Uma equipe de desenvolvimento de jogos está criando um sistema de cadastro de jogadores que utiliza strings para armazenar nomes de usuário. Durante os testes, foi descoberto que alguns nomes não estavam sendo armazenados corretamente quando continham espaços em branco, como "João Silva" ou "Ana Carolina". O programador júnior relatou que estava usando a função `scanf()` para ler os nomes, mas estava ocorrendo um comportamento inesperado onde apenas a primeira palavra era capturada. O líder técnico explicou que existem diferenças importantes entre as funções de entrada de dados para strings em C, e que a escolha da função adequada depende do tipo de entrada que se espera receber do usuário.

Considerando as informações apresentadas, avalie as asserções a seguir e a relação proposta entre elas:

- I. A função `scanf()` com `%s` interrompe a leitura quando encontra um espaço em branco, sendo inadequada para strings que contenham espaços.

PORQUE

- II. A função `gets()` lê caracteres até encontrar o caractere de nova linha (`\n`), permitindo a leitura de strings com espaços em branco.

A respeito dessas asserções, assinale a opção correta.

**RESPOSTA:** As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.

### 6ª QUESTÃO

O uso combinado de funções e arquivos em C permite criar programas robustos e organizados, em que tarefas específicas, como salvar dados, ler registros ou processar informações, são encapsuladas em funções dedicadas, melhorando a manutenibilidade e clareza do código.

Considerando o texto apresentado, sobre esses conceitos, analise as afirmações a seguir:

- I. Uma função é um bloco de código que pode receber parâmetros, executar instruções e retornar um valor opcional.
- II. Arquivos em C permitem armazenar dados de forma permanente, podendo ser manipulados em modos como "r" (leitura), "w" (escrita) e "a" (acréscimo).
- III. A função `fopen(nomeArquivo, modo)` abre um arquivo e retorna um ponteiro do tipo `FILE*`, que é usado nas demais operações de leitura e escrita.
- IV. Uma função que escreve em arquivo precisa necessariamente retornar um valor inteiro, caso contrário não é considerada válida em C.

É correto o que se afirma em:

**RESPOSTA:** I, II e III, apenas.

### 7ª QUESTÃO

Em C, é comum combinar matrizes (arrays multidimensionais) e structs para organizar dados complexos.

Considerando o texto apresentado, sobre esses conceitos, analise as afirmações a seguir:

- I. Uma matriz é um array de arrays, permitindo armazenar elementos em linhas e colunas, sendo acessada pela sintaxe `matriz[i][j]`.
- II. Uma struct é um tipo de dado que permite agrupar variáveis de diferentes tipos sob um mesmo nome.
- III. É possível declarar uma matriz de structs, permitindo organizar múltiplos registros bidimensionais de forma estruturada.
- IV. Cada elemento de uma struct que está dentro de uma matriz de structs só pode ser acessado usando a função `printf`, não podendo ser acessado diretamente.

É correto o que se afirma em:

**RESPOSTA:** I, II e III, apenas.

### 8ª QUESTÃO

Analise a seguinte função elaborada na linguagem C.

```
int func_f(int n) {  
    int a, b;  
    if (n == 0) {  
        return(1);  
    }  
    a = n - 1;  
    b = func_f(a);  
    return(n*b);  
}
```

Caso, no programa principal, seja colocada a linha `printf("%d", func_f(4));` para chamada da função o valor impresso na saída padrão será:

**RESPOSTA:** 24**9ª QUESTÃO**

Durante o desenvolvimento de um sistema de processamento de texto, uma equipe de programadores precisa implementar uma funcionalidade que conte o número de palavras em um documento. O sistema deve ler um arquivo texto linha por linha até encontrar o final do arquivo (EOF), processar cada linha para contar as palavras e manter um contador total. O arquiteto de software sugeriu o uso de funções específicas para manipulação de arquivos em C, destacando a importância de verificar se o arquivo foi aberto corretamente antes de iniciar qualquer operação de leitura. A equipe está analisando as melhores práticas para manipulação de arquivos, incluindo abertura, verificação de erros e fechamento adequado.

Considerando as informações apresentadas, avalie as asserções a seguir e a relação proposta entre elas:

I. A função `fopen()` retorna `NULL` quando ocorre erro na abertura do arquivo, sendo essencial verificar este retorno antes de realizar operações de leitura.

PORQUE

II. A verificação de erro na abertura de arquivo é uma prática opcional em C, pois o programa continuará funcionando mesmo com arquivos inexistentes.

A respeito dessas asserções, assinale a opção correta.

**RESPOSTA:** A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.**10ª QUESTÃO**

Durante uma aula de laboratório de programação, os estudantes estão aprendendo sobre operadores aritméticos em C. O professor apresenta um exercício onde é necessário calcular a média de três notas e exibir o resultado com duas casas decimais. Um estudante pergunta sobre a diferença entre os operadores de divisão (`/`) quando aplicados a números inteiros versus números reais (`float`). O professor explica que o comportamento do operador de divisão pode variar dependendo dos tipos de dados envolvidos na operação, o que pode afetar significativamente o resultado final do cálculo, especialmente em situações onde a precisão decimal é importante.

Considerando as informações apresentadas, avalie as asserções a seguir e a relação proposta entre elas:

I. O operador de divisão (`/`) entre dois números inteiros em C resulta em uma divisão inteira, descartando a parte fracionária.

PORQUE

II. Para obter resultado com casas decimais, pelo menos um dos operandos deve ser do tipo `float` ou `double`.

A respeito dessas asserções, assinale a opção correta.

<b>RESPOSTA:</b>	As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
------------------	---