

## Protocolo de Finalização Nº 000100283907

O aluno Gultembergue Carlos com RA **25531299-5** finalizou a atividade **ATIVIDADE 2 - ADSIS - LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO - 54\_2025** em **25/11/2025 21:23:26**

### ATIVIDADE 2 - 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 linguagem C foi criada na década de 1970 por Dennis Ritchie nos Laboratórios Bell e se tornou uma das linguagens mais utilizadas no desenvolvimento de sistemas e aplicações. Sua principal característica é combinar a eficiência da linguagem de montagem com a simplicidade de uma linguagem de alto nível, permitindo a criação de programas portáteis, que podem ser executados em diferentes máquinas. No entanto, essa liberdade pode ser um desafio para programadores iniciantes, já que construções incorretas podem ser aceitas pelo compilador. A estrutura básica de um programa em C inclui a função `main()`, que é sempre o ponto inicial de execução, e geralmente faz uso de bibliotecas como a `stdio.h`, responsável por operações de entrada e saída de dados.

Com base no texto apresentado e nos conceitos iniciais da linguagem C, assinale a alternativa correta:

**RESPOSTA:** O arquivo de cabeçalho `stdio.h` é utilizado para operações de entrada e saída, como `printf` e `scanf`.

#### 2ª QUESTÃO

A empresa FinanceControl está desenvolvendo um módulo de cálculo de juros compostos que deve ser utilizado em diferentes partes do sistema. O desenvolvedor senior decidiu criar uma função reutilizável que receba como parâmetros o capital inicial, a taxa de juros e o tempo, retornando o montante final. Durante a implementação, surgiu uma questão sobre a passagem de parâmetros: alguns valores precisam ser preservados após a execução da função (como o capital inicial para relatórios), enquanto outros podem ser modificados durante os cálculos internos da função. A equipe técnica está debatendo entre utilizar passagem por valor ou por referência, considerando que a função deve ser eficiente e não deve alterar acidentalmente os valores originais das variáveis principais do programa.

Com base no texto apresentado, analisando a situação e os conceitos de passagem de parâmetros em C, qual abordagem é mais adequada?

**RESPOSTA:** Passagem por valor para os parâmetros de entrada e retorno do resultado calculado.

#### 3ª QUESTÃO

Em C, as estruturas condicionais permitem desviar o fluxo de execução de acordo com uma condição lógica. O comando `if` avalia uma expressão e, se verdadeira, executa um bloco de instruções. Já o `if...else` possibilita escolher entre dois caminhos: um quando a condição é verdadeira e outro quando é falsa. Para tratar mais de duas situações, podemos encadear condições usando `if...else if...else`.

Considere o seguinte trecho de código em C:

```
c

int valor = 15;

if (valor < 20) {
    printf("Saque mínimo permitido: R$ 20,00\n");
} else if (valor > 1000) {
    printf("Limite máximo de saque excedido\n");
} else {
    printf("Saque autorizado\n");
}
```

Imagem 1: Trecho de código em C

Fonte: o autor.

Com base no texto e no trecho do código apresentado, qual será a saída exibida na tela?

<b>RESPOSTA:</b>	Saque mínimo permitido: R\$ 20,00.
------------------	------------------------------------

#### 4ª QUESTÃO

Na linguagem C, variáveis são espaços de memória capazes de armazenar valores que podem mudar durante a execução do programa, enquanto constantes guardam valores fixos, definidos pelo programador e que não se alteram. As expressões combinam variáveis, constantes e operadores para produzir resultados, como em fórmulas matemáticas. A linguagem oferece operadores aritméticos (`+`, `-`, `*`, `/`, `%`), relacionais (`==`, `!=`, `<`, `>`, `<=`, `>=`) e lógicos (`&&`, `||`, `!`). Além disso, o comando de atribuição `=` é utilizado para armazenar valores em variáveis.

Considerando o texto apresentado, sobre variáveis, constantes, expressões e operadores em C, avalie as afirmações a seguir.

- I. Variáveis armazenam valores que podem ser alterados durante a execução do programa, enquanto constantes mantêm valores fixos.
- II. O operador `=` é utilizado para comparar dois valores, resultando em verdadeiro ou falso.
- III. A expressão `media = (n1 + n2 + n3) / 3;` combina variáveis, operadores e atribuição.
- IV. O operador `%` é usado para obter o resto da divisão entre dois números inteiros.

É correto o que se afirma em:

<b>RESPOSTA:</b>	I, III e IV, apenas.
------------------	----------------------

## 5ª QUESTÃO

Analise o algoritmo a seguir:

```
#include <stdio.h>
#define MAX 5

int pertence(int x, int L[], int n);
int main(void){
    int A[MAX] = {9, 1, 2, 7};
    int i, numero;
    numero = 1;
    if(i = pertence(numero, A, MAX))
        printf("Sim!!! O valor pertence a posição [%d]", i);
    else
        printf("Não!!!!");
}
int pertence(int x, int L[], int n) {
    int i, j;
    j = 1;
    for(i=0; i<n; i++)
        if( x == L[i] )
            return i;
    return 0;
}
```

De acordo com o algoritmo, assinale a alternativa correta:

- I – A função pertence() devolve o valor encontrado, que no caso é 1 e não a posição.
- II – A função pertence() devolve o índice da posição do valor encontrado, que é a posição 1 do vetor.
- III – O vetor A inicia da posição 0.
- IV – O vetor A inicia da posição 1.

É correto o que se afirma em:

<b>RESPOSTA:</b>	II e III, apenas.
------------------	-------------------

## 6ª QUESTÃO

Analise o código abaixo.

```
#include<stdio.h>
#include<string.h>
#define NIVEL_MINIMO_CAIXA 10
#define NIVEL_MAXIMO_CAIXA 980

int main(void)
{
    float nivelMinimoCaixa, nivelMaximoCaixa, nivelMedicao;
    printf("Informe o nível da medicao: ");
    scanf("%f", &nivelMedicao);
    char mensagem[30];

    if (NIVEL_MINIMO_CAIXA >= nivelMedicao){
        mensagem = "Ligar Bomba";
    }else if (nivelMedicao >= NIVEL_MAXIMO_CAIXA){
        mensagem = "Desligar Bomba";
    }else{
        mensagem = "Manter Bomba ligada";
    }
    printf("%s", mensagem);
}
```

De acordo com o contexto apresentado acima, assinale a alternativa correta que corrige os erros do código a fim de que seja impresso na tela o valor da variável mensagem.

**RESPOSTA:** Será apresentado erro de incompatibilidade de tipos na atribuição das mensagens na variável mensagem. O correto é utilizar a função strcpy();

## 7ª QUESTÃO

As funções intrínsecas da linguagem C permitem realizar cálculos matemáticos prontos, como `sqrt(x)` para raiz quadrada e `pow(x,y)` para potência. Já a entrada de dados pode ser feita com `scanf`, que armazena valores digitados pelo usuário em variáveis. Por fim, a saída é realizada, em geral, pela função `printf`, que permite mostrar mensagens e resultados formatados.

Um professor de programação pediu que os alunos construíssem um programa em C que:

Leia um número real positivo digitado pelo usuário.

Calcule a raiz quadrada desse número utilizando uma função intrínseca.

Mostre na tela o valor lido e o resultado do cálculo.

Durante a explicação, quatro alunos deram opiniões diferentes sobre como resolver o problema

Considerando o texto e as condições apresentadas, sobre a linguagem C, avalie as afirmações a seguir.

- I. Para ler o número, pode-se usar `scanf("%f", &num);`, armazenando-o em uma variável do tipo `float`.
- II. Para calcular a raiz quadrada, a função correta é `pow(num, 1/2)`.
- III. O comando de saída deve ser algo como `printf("Raiz: %.2f", sqrt(num));`.
- IV. Para usar a função `sqrt`, é necessário incluir a biblioteca `math.h`.

É correto o que se afirma em:

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

## 8ª QUESTÃO

Uma software house está desenvolvendo um aplicativo de gerenciamento de biblioteca que precisa organizar informações sobre livros. Cada livro possui múltiplos atributos: código (inteiro), título (string), autor (string), ano (inteiro), preço (float) e disponibilidade (char). O analista responsável pelo projeto precisa decidir entre usar múltiplas variáveis separadas ou uma estrutura mais organizada para representar essas informações. Considerando que o sistema precisará gerenciar centenas de livros e realizar operações como busca, ordenação e relatórios, a escolha da estrutura de dados adequada impactará diretamente na manutenibilidade e eficiência do código.

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

I. A utilização de struct é inadequada para esta situação, pois impede o acesso individual aos campos de cada livro.

PORQUE

II. Structures (structs) em C permitem agrupar informações relacionadas de tipos diferentes em uma única estrutura, facilitando a organização e manipulação dos dados.

A respeito dessas asserções, assinale a opção correta.

<b>RESPOSTA:</b>	A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
------------------	--

## 9ª QUESTÃO

Durante uma aula prática de programação, os estudantes estão aprendendo sobre a estrutura condicional simples (if) em C. O professor apresenta um exercício onde é necessário verificar se um número digitado pelo usuário é par e, caso seja, exibir sua raiz quadrada na tela. Um estudante questiona sobre a necessidade das chaves {} na estrutura if, especialmente quando há apenas uma instrução a ser executada. O professor explica que existem regras específicas para o uso das chaves em estruturas condicionais, relacionadas à quantidade de comandos que serão executados dentro do bloco condicional.

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

I. Em C, é obrigatório o uso de chaves quando existe mais de um comando a ser executado dentro da estrutura condicional if.

PORQUE

II. As chaves delimitam um novo bloco de comandos, definindo claramente quais instruções pertencem à estrutura condicional.

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.
------------------	---

## 10ª QUESTÃO

Analise o programa em C

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int y = 15;
```

```
    int z = 15;
```

```
    while (y >= 0){
```

```
        y--;
```

```
        printf("While ");
```

```
}
```

```
    do{
```

```
        printf("Do While ");
```

```
        z--;
```

```
}while(z > 0);
```

```
}
```

Assinale a alternativa que representa a quantidade de vezes que é exibido em tela a palavra "While" e "Do While" respectivamente.

**RESPOSTA:** 16 vezes para "While" e 16 vezes para "Do While"