

Discente: _____ Matrícula: _____

01. Avalie as seguintes sentenças sobre vetores e strings em C e escolha a alternativa correta.

- I. É possível utilizar a notação de colchetes [] para acessar caracteres individuais em uma string.
 - II. Um vetor em C é uma coleção de elementos de tipos iguais, armazenados sequencialmente na memória.
 - III. A função `strcmp()` retorna 1 se duas strings forem iguais.
 - IV. É possível acessar elementos de um vetor utilizando um índice fora dos limites do vetor, mas isso pode levar a resultados inesperados.
- a. () Apenas II está correta.
 - b. () Apenas I, II e IV estão corretas.
 - c. () Apenas I, IV e III estão corretas.
 - d. () Apenas III está correta.

02. Qual das seguintes afirmações sobre estruturas (`struct`) em C é FALSA?

- a. () Estruturas permitem agrupar dados de diferentes tipos em uma única entidade.
- b. () Uma estrutura pode conter membros que são outras estruturas, criando estruturas aninhadas.
- c. () Estruturas em C são alocadas dinamicamente na memória, por padrão.
- d. () O operador ponto (.) é usado para acessar membros de uma estrutura.

03. Em relação a ponteiros em C, qual a principal função de um ponteiro e qual operador é utilizado para obter o endereço de memória de uma variável, respectivamente?

- a. () Armazenar o endereço de memória de uma variável, e o operador é o * (asterisco).
- b. () Apontar para a próxima instrução do programa, e o operador é a -> (seta).
- c. () Armazenar o endereço de memória de uma variável, e o operador é o & (e comercial).
- d. () Alocar memória dinamicamente, e o operador é a -> (seta).

04. Considere a declaração de estruturas aninhadas em C:

```
struct endereco {  
    char rua[50];  
    int numero;  
};
```

```
typedef struct pessoa {  
    char nome[50];  
    int idade;  
    struct endereco end;  
} Pessoa;
```

Como você acessaria o número da casa da variável `pessoa_01` do tipo `Pessoa`? Justifique sua resposta.

- a. () `pessoa_01.end.numero`
- b. () `pessoa_01->end->numero`
- c. () `pessoa_01.numero`
- d. () `endereco.numero`

05. Informe qual a saída do trecho de código a seguir. Justifique sua resposta.

```
int arr[] = {100, 200, 300};  
int *ptr = arr;  
  
*ptr += 50;  
ptr++;  
ptr++;  
*ptr += 100;  
  
for(int i = 0; i<3; i++){  
    printf("%d ", arr[i]);  
}
```

- a. () 52 100 200
- b. () 150 200 400
- c. () 100 200 300
- d. () 150 202 400

06. Informe qual a saída do trecho de código a seguir. Justifique sua resposta.

```
int x = 5;
int *ptr1 = &x;
int **ptr2 = &ptr1;
**ptr2 += 1;

printf("%d", **ptr2);
```

- ☐ Endereço incrementado de x
- ☐ Endereço de ptr1 incrementado em 1 unidade
- ☐ 5
- ☐ 6

07. Qual das seguintes afirmações sobre o operador sizeof() é verdadeira em relação à alocação de memória em C?

- ☐ Retorna o tamanho em bytes de um tipo de dado ou variável.
- ☐ Retorna o número de elementos em um array dinamicamente alocado.
- ☐ Retorna o endereço de memória de uma variável dinamicamente alocada.
- ☐ Aloca um espaço de memória do "tamanho de" bytes.

08. Qual função é utilizada para desalocar um bloco de memória previamente alocado em C?

- ☐ free()
- ☐ system("free")
- ☐ calloc(0)
- ☐ freealloc()

09. Avalie o trecho de código abaixo e marque a alternativa que descreve corretamente o seu comportamento. Justifique sua resposta.

```
int *ptr = (int*) malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++) {
    *(ptr + i) = num;
}
free(ptr);
```

- ☐ O código aloca memória para 5 inteiros, inicializa os valores com o dobro de cada número lido e libera a memória.

- ☐ O código aloca memória para um vetor de 5 inteiros, mas apresenta um erro por variável não declarada.
- ☐ O código aloca memória para 20 inteiros, inicializa os valores com o i-ésimo valor da posição de memória de cada elemento do vetor.
- ☐ O código não libera a memória alocada, pois a memória é liberada com o uso coringa de realloc(ptr, 0);

- Nomeie cada arquivo como ex10 e ex11..
- Comprima os arquivos e nomeie o zip como matricula_primeiroNome_revisaoAED1.
Exemplo:
202400100_magravildes_revisaoAED1.zip
- Submeta o arquivo no local indicado.

10. Implemente um programa que receba o nome completo, matrícula, anos trabalhados e departamento (nome, quantidade de funcionários, anos de existência) de uma quantidade fixa de funcionários de uma empresa, informada pelo usuário. Em seguida imprima na tela os dados das estruturas aninhadas lidas. Durante a listagem, caso um funcionário possua anos trabalhados maiores ou iguais a 30 anos, indique que ele(a) já pode solicitar a aposentadoria. OBS: não deve ser usado ponteiro, nem alocação dinâmica neste exercício.

11. Escreva um programa em C que utilize alocação dinâmica de memória para criar uma matriz de inteiros, triangular de tamanho M, informado pelo usuário. Sabe-se que a cada nível a quantidade de colunas é incrementada até atingir o valor M na última linha. Durante o processo de preenchimento, cada célula deve armazenar o quíntuplo do valor digitado pelo usuário. Ao final, usando a mesma matriz alocada, exiba a matriz original e quintuplicada. Lembre-se de que como é uma matriz dinamicamente alocada, os elementos precisam ter seus espaços de memória liberados adequadamente.