

ATIVIDADE 2 - ESOFT - ESTRUTURA DE DADOS I - 52/2023**Período:**29/05/2023 08:00 a 16/06/2023 23:59 (Horário de Brasília)**Status:**ENCERRADO**Nota máxima:**0,50**Gabarito:**Gabarito será liberado no dia 17/06/2023 00:00 (Horário de Brasília)**Nota obtida:**0,50**1ª QUESTÃO**

Considere o excerto de código a seguir:

```
void remover() {  
    if (estr.ini == estr.fim) {  
        printf("\nA estrutura está vazia, não há nada para remover!\n\n");  
        system("pause");  
    }else {  
        int i;  
        for (i = 0; i < tamanho; i++) {  
            estr.dados[i] = estr.dados[i+1];  
        }  
        estr.dados[filas.fim] = 0;  
        estr.fim--;  
    }  
}
```

O algoritmo representa a remoção de elementos de um determinado tipo de estrutura de dados.

Assinale a alternativa que indica o tipo de estrutura que o algoritmo se refere.

ALTERNATIVAS

- ☒ Fila
- ☐ Lista
- ☐ Pilha
- ☐ Heap
- ☐ Grafo

Atenção! Questão anulada.

ALTERNATIVAS

- ☐ II, apenas.
- ☐ III, apenas.
- ☐ I e II, apenas.
- ☐ II e III, apenas.
- ☒ I, II e III.

Atenção! Questão anulada.

ALTERNATIVAS

☐ `int main(void) {
 struct funcionario func;
 printf("Nome: ");
 gets(func.nome);
 printf("%s", func.nome);
}`

☐ `int main(void) {
 struct funcionario func;
 printf("Nome: ");
 puts(func.nome);
 printf("%s", func.nome);
}`

☐ `int main(void) {
 struct funcionario func;
 printf("Nome: ");
 scanf("%s", func.nome);
 printf("%s", func.nome);
}`

☐ `int main(void) {
 struct funcionario func;
 printf("Nome: ");
 scanf("%s", &func.nome);
 printf("%s", func.nome);
}`

☐ `int main(void) {
 struct funcionario func;
 printf("Nome: ");
 getchar(func.nome);
 printf("%s", func.nome);
}`

4ª QUESTÃO

Pense em um vetor como uma matriz de uma única linha e quantidade de colunas equivalente ao seu tamanho. O vetor é uma estrutura homogênea, por isso só pode armazenar um único tipo de dado.

(Fonte: OLIVEIRA, Pietro Martins de; Pereira, Rogério de Leon. Estrutura de Dados I. Maringá-Pr.: Unicesumar, 2019.)

Na condição de programador(a) o seu trabalho é prover verificação dos limites onde for necessário. Nesse contexto, quatro programadores analisaram o código abaixo.

```
#include <stdio.h>

int main(void) {
    int v[10], i;
    for (i = 0; i < 100; i++)
        v[i] = i;
}
```

Cada um dos quatro programadores chegaram a uma conclusão:

- I. O código compilará e está correto.
- II. O código compilará, mas está incorreto.
- III. O código não compilará e está incorreto.
- IV. O código não compilará, mas está correto.

Está(ão) correto(s) o(s) programador(es).

ALTERNATIVAS

- ☐ I, apenas.
- ☒ II, apenas.
- ☐ III, apenas.
- ☐ IV, apenas.
- ☐ I, II, III e IV.

5ª QUESTÃO

Um registro é uma coleção de variáveis e permite o armazenamento de informações heterogêneas, ou seja, de tipos diferentes. Um registro possibilita ao programador criar tipos de dados específicos e personalizados. A declaração de um registro se dá pela palavra reservada *struct*, seguida pelo conjunto de elementos que o compõem.

(Fonte: OLIVEIRA, Pietro Martins de; Pereira, Rogério de Leon. Estrutura de Dados I. Maringá-Pr.: Unicesumar, 2019).

Observe a estrutura abaixo:

```
struct logradouro{  
    char endereco[30];  
    int numero;  
    char complemento[10];  
    char cidade[20];  
    char estado[2];  
    unsigned int cep;  
};
```

Assinale a alternativa correta para declarar a variável info do tipo logradouro:

ALTERNATIVAS

- ☒ struct logradouro info;
- ☐ struct info logradouro;
- ☐ info = logradouro;
- ☐ logradouro info;
- ☐ info logradouro;