

Estrutura de dados

Tema 2 – Estrutura de dados heterogêneas

1. Para melhorar a performance de um jogo, os desenvolvedores precisam armazenar e manipular eficientemente as posições dos personagens em um mundo virtual tridimensional. Qual tipo de dado em C é mais adequado para representar as posições tridimensionais dos personagens?
 - a. Estrutura contendo três floats.
2. Sobre estruturas de dados, assinale a alternativa CORRETA.
 - a. Pilhas são tipos de dados abstratos caracterizadas pela política “primeiro a entrar, último a sair”.
3. A sigla FIFO refere-se a estruturas de dados do tipo fila. Como é o funcionamento em uma FIFO?
 - a. O primeiro objeto inserido na fila é também o primeiro a ser removido.
4. Em um projeto de software para gerenciamento de bibliotecas, utiliza-se a estrutura de dados heterogênea para modelar livros. Cada livro é representado por uma estrutura contendo título, autor, ano de publicação e status de empréstimo. Qual estrutura de dados em C é mais apropriada para armazenar os detalhes de cada livro no projeto descrito?
 - a. Estrutura (struct)
5. Em um sistema de gerenciamento de estoque, utiliza-se uma estrutura para armazenar informações dos produtos, com código, descrição, quantidade e preço. É necessário acessar rapidamente essas informações para atualizações frequentes. Como se deve acessar os membros de uma estrutura em C para realizar atualizações?
 - a. Operador ponto(.).

6. Considere o programa a seguir, elaborando na linguagem ANSI C:

```
#include <stdio.h>

Int* pa, a;
Int main() {a = 10; pa= &a; printf("%d",a); printf("%d", *pa); }
```

Assinale a alternativa correta sobre a saída (output) obtida ao se executar o programa:

- a. 8 8
7. A linguagem C permite alocar (reservar) dinamicamente (em tempo de execução) blocos de memórias utilizando ponteiros. A esse processo dá-se o nome de alocação dinâmica, que faz uso das funções malloc, calloc, realloc e free disponíveis na biblioteca stdlib.h. Para liberar um bloco de memória previamente alocado, por meio de um único parâmetro de entrada, faz-se uso de qual função?
- a. Free

Tema 3 – Modularização

1. Observe o trecho abaixo, escrito na linguagem C.

```
void imprimecabecalho(){  
    ...  
}  
void calcula(){  
    int soma;  
    ...  
    imprimecabecalho();  
}
```

2. Com base nesse código, é correto afirmar que:
 - a. O tempo de vida da variável soma estende-se durante o tempo em que a função imprimecabecalho() é executada.

3. Considere o código a seguir escrito na linguagem C.

```
#include  
int main(){  
    printf("Valor total: %f\n", 9,1415169265);  
    return(0);  
}
```

Assinale a alternativa que representa a saída correta.

- a. Valor total: 9.1
4. A modularização é importante para organizar melhor o código, facilitar a manutenção, entre outras coisas. Sobre a modularização, assinale a alternativa correta.
 - a. É um método que consiste em decompor um programa em uma série de subprogramas individuais.

5. As sub-rotinas são procedimentos e funções da programação modularizada. Um parâmetro passado por valor para uma sub-rotina se comportará como uma variável local, isto é, qualquer modificação no valor desta variável não será visível fora da sub-rotina. Sobre a passagem de parâmetros, analise as afirmativas abaixo.
- I – Na passagem por valor ocorre uma troca de valores entre variáveis na sub-rotina.
 - II – Na passagem por valor ocorre uma cópia do valor da variável em todas as posições de memória, sendo possível alterar o conteúdo da variável original a qualquer momento.
 - III. A passagem por referência nunca deve ser utilizada.
 - a. Apenas a afirmativa I está correta
6. Na linguagem C, o cabeçalho de biblioteca `stdio.h` possui definições de sub-rotinas relativas às operações de entrada/saída padrão. Uma das funções tratadas na biblioteca, associada a esse cabeçalho, é a função.
- a. `Printf`
7. Essas variáveis que são declaradas fora da sub-rotina, podendo ser acessíveis em todos os escopos, em qualquer lugar de um programa, disponíveis durante toda a execução do programa. Marque a alternativa que apresenta o tipo de variável descrito acima.
- a. Global
8. Há duas maneiras de se passar argumentos ou parâmetros para funções: por valor e por referência. Sobre passagem de parâmetros, analise as seguintes afirmativas:
- I. Na passagem por referência, o que é passado como argumento no parâmetro formal é o endereço da variável.
 - II. Na passagem por valor, o valor é copiado do argumento para o parâmetro formal da função.
 - III. Por exemplo, quando duas variáveis inteiras `i1` e `i2` são passadas por valor à função `troca()` chamada pelo programa principal, elas também são alteradas no programa principal.
 - IV. Na passagem por referência, dentro da função, o argumento real utilizado na chamada é acessado através do seu endereço, sendo assim alterado.
 - V. Na passagem por valor, quaisquer alterações feitas nestes parâmetros dentro da função não irão afetar as variáveis usadas como argumentos para chamá-la.
- Está CORRETO o que se afirma em:

a. I, II, IV e V, apenas

Tema 4 – Listas, pilhas e filas

1. Sobre listas duplamente encadeadas, afirma-se:
 - I) Cada nó usa o dobro do número de campos ponteiro de uma lista simplesmente encadeada.
 - II) A complexidade de remoção é metade da complexidade de remoção em lista simplesmente encadeada.
 - III) Não permitem a inserção de nó no meio da lista.É correto apenas:
 - a. II.
2. O acesso ao elemento de uma estrutura de dados tipo pilha se restringe ao mais recente na pilha. Já o acesso a um elemento de uma estrutura tipo fila ocorre ao dado há mais tempo na fila. Sobre pilhas e filas, avalie as assertivas a seguir:
 - I – Uma forma de evitar o desperdício de memória numa fila em alocação sequencial é utilizar-se lista circular.
 - II – Em uma pilha em alocação encadeada, a complexidade de remoção é $O(n)$.
 - III – Pilhas têm a propriedade de inverter a ordem de cadeias, enquanto as filas mantêm a ordem.A opção que contém todas as assertivas corretas é:
 - a. I e II.
3. Assinale, das alternativas abaixo, a única que identifica respectivamente uma Estrutura de Dados do tipo FIFO (First in, First Out) e uma outra com a estrutura de dados do tipo LIFO (Last In, First Out):
 - a. Fila, Pilha
4. Uma lista ordenada alocada sequencialmente possui como desvantagem:
 - a. Tamanho limitado de memória alocada para lista.
5. Uma estrutura de dados onde existe uma coleção ordenada de entidades sendo a metodologia de busca com base no deslocamento relativo ao primeiro (cabeça) da coleção chama-se.
 - a. Lista
6. Várias estruturas de dados podem ser utilizadas para armazenar dados de uma aplicação. Em relação ao assunto, assinale a alternativa correta.

- a. A estrutura de dados tipo fila utiliza a ideia do primeiro a ser inserido, será o primeiro a ser retirado
7. Insira os dados de entrada numa fila. Em seguida, retire cada dado da fila e insira numa pilha. Mostre a pilha. Depois retire os dados da pilha e insira na fila. Mostre a fila.

Dados de entrada: 11, 12, 23, 14, 25, 50, 8, 18, 29, 10

As estruturas mostradas ficam

I. pilha: (topo) 10 – 29 – 18 – 8 – 50 – 25 – 14 – 23 – 12 – 11

II. Fila (começo) 11 – 12 – 23 – 14 – 25 – 50 – 8 – 18 – 29 – 10 (fim)

III. Fila (começo) 10 – 29 – 18 – 8 – 50 – 25 – 14 – 23 – 12 – 11 (fim)

IV. Pilha: (topo) 11 – 12 – 23 – 14 – 25 – 50 – 8 – 18 – 29 – 10

V. A fila mostrada fica com os elementos em ordem invertida dos dados de entrada

Está correto o que se afirma APENAS em:

- a. I, III e V.

Tema 5 – Ordenação

1. Todos os algoritmos de ordenação interna devem ter complexidade de espaço de
 - a. $O(n)$
2. Algoritmos de ordenação são responsáveis por ordenar elementos de uma estrutura de dados de forma completa ou parcial. Sobre a complexidade dos algoritmos de ordenação, assinale, a seguir, o algoritmo de ordenação que, no pior caso, tem complexidade igual a $O(n \log n)$
 - a. Merge sort
3. A ordenação é um dos conceitos fundamentais da ciência da computação e desempenha um papel crucial em várias áreas, como algoritmos, estruturas de dados e processamento de dados em geral. Qual das alternativas abaixo indica um algoritmo de ordenação?
 - a. Bubble sort
4. A ordenação por seleção é um algoritmo simples e intuitivo. Na ordenação por seleção, é necessário permutar o vetor. Para esse caso, o significado de ‘permutar’ é
 - a. Rearranjar os elementos do vetor
5. Algoritmos de ordenação baseados em comparação entre elementos da sequência tem complexidade computacional mínima de:
 - a. $O(n \log n)$
6. Considere utilizar o algoritmo Bubble Sort para ordenar, em ordem crescente, a sequência de números: 17, 42, 37, 8, 77, 52, 25. Se a sequência original for a iteração zero, qual será a sequência de números da segunda iteração?
 - a. 17, 31, 8, 37, 43, 25, 52, 77
7. Algoritmos de ordenação são responsáveis por ordenar elementos de uma estrutura de dados de forma completa ou parcial. Sobre a complexidade dos algoritmos de ordenação, assinale a seguir o algoritmo de ordenação que, no pior caso, tem a complexidade igual a $O(n \log n)$.
 - a. Merge Sort

8. Leia o seguinte trecho de código fonte escrito em java

```
// laço com a quantidade de elementos do vetor (n=10)
For(n = 1; n <=10; n++){
    for(i = 0; i <= 8; i++) {
        if(vet[i]>vet[i+1]){
            aux = vet[i];
            vet[i]=vet[i+1];
            vet[i+1]=aux;
        }
    }
}
```

O algoritmo de ordenação implementado nesse trecho de código-fonte é o

- a. Bubble Sort

9. Considere uma estrutura de dados do tipo vetor de tamanho 6 (seis) inicializado com as chaves abaixo:

	1	2	3	4	5	6
vetor	[79]	[82]	[68]	[69]	[78]	[65]
passo1	[65]	[82]	[68]	[69]	[78]	[79]
passo2	[65]	[68]	[82]	[69]	[78]	[79]
passo3	[65]	[68]	[69]	[82]	[78]	[79]
passo4	[65]	[68]	[69]	[78]	[82]	[79]
passo5	[65]	[68]	[69]	[78]	[79]	[82]

O método apresentado é:

- a. Ordenação por Seleção