

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