



RESPOSTAS DE DÚVIDAS ENVIADAS NA AULA AO VIVO 1

(APENAS AS NÃO RESPONDIDAS NA AULA AO VIVO)

Dúvidas sobre o conteúdo da aula:

1. Linguagem C é mais utilizada em front ou back-end?

a. Resposta:

- i. É mais utilizada no **Back-End**, e não é comum no Front-End do desenvolvimento web. Vamos lembrar que C é uma linguagem de baixo nível, muito mais próxima do hardware, sendo ideal para sistemas onde o controle direto de recursos e eficiência são a prioridade.

2. No VS Code, qual a tecla para ir salvando?

a. Resposta:

- i. ctrl+s

3. Quando digita case 1: o break vem automaticamente?

a. Resposta:

- i. Essa opção de autocompletar código depende da interface de desenvolvimento (IDE) que utiliza. É uma função ótima para programadores experientes, mas é recomendável que programadores iniciantes tenham a prática de digitar os comandos para poder memorizar os comandos utilizados e sua estrutura. Vamos codar para criar uma memória muscular! Programadores não podem ficar dependentes de IDEs. O professor pode dar exemplos na aula ao vivo apenas para conhecerem as possibilidades!

4. Especificador inteiro: %d

a. Resposta:

- i. Apenas complementando, o %d especifica o formato de inteiro decimal. Podemos também especificar inteiros com o %i. Enquanto %d especifica necessariamente base 10 (decimal), o %i pode especificar em formato octal (base 8) e hexadecimal (base 16), lendo automaticamente a base, a partir do formato. Nota: Para inserir um valor em formato hexadecimal, o valor deve ser fornecido precedendo "0x" e o valor em formato octal, o valor deve ser fornecido precedendo "0".



Outras dúvidas:

5. O que é linguagem de montagem (linguagem de máquina ou assembly)?

a. Resposta:

- i. **Linguagem de Máquina** é o código binário que é diretamente entendido pelo processador. Uma cadeia de 0s e 1s com uma ordem de instruções a serem executadas.
- ii. **Linguagem de Montagem** ou **Assembly** são sinônimos para um tipo de codificação de baixo nível (ou seja, bem próxima à Linguagem de Máquina), feita por meio de **mnemônicos** (abreviações) que representam as instruções a serem executadas pelo processador do computador. As instruções Assembly variam para cada tipo de processador (x86, ARM, entre outros).
- iii. Assembly está bem próxima da Linguagem de Máquina, mas ainda sendo possível a leitura por humanos, embora não tão amigável como linguagens de programação de alto nível.
- iv. Por ser muito direta e próxima da máquina, tem vantagens em ser muito mais eficiente, sendo amplamente utilizada em áreas onde impera a otimização e controle, como o desenvolvimento de sistemas operacionais e firmwares.

6. O que é operação aritmética sobre ponteiro, operação sobre ponteiro e operação sobre a palavra binária?

a. Resposta:

- i. **Operação aritmética sobre ponteiro** (como + e -): Ajustam o ponteiro em relação ao tipo de dado apontado, permitindo navegar por elementos de arrays ou blocos de memória.
- ii. **Operação sobre ponteiro**: Envolve manipular o valor do ponteiro em si (o endereço) ou o conteúdo do endereço que ele aponta, usando operadores como * (desreferenciação) e & (obtenção de endereço).
- iii. **Operação sobre palavra binária**: É a manipulação direta de bits de uma "palavra" (quantidade de bits processada de uma vez pela CPU, como 32 ou 64 bits) usando operações bit a bit (&, |, ^, ~, deslocamento <<, >>). É usada para otimizar cálculos ou controlar hardware em baixo nível.

7. Um exemplo de utilidade do tipo void.

a. Resposta:

- i. Os principais usos do tipo void são em funções que não retornam nenhum valor ou em ponteiros genéricos (void *), que não apontam para nenhum tipo específico de dado). Os ponteiros genéricos são úteis em funções que manipulam tipos diferentes

de dados. É usando, por exemplo, na função padrão *malloc()*, utilizada para alocação de memória, retornando um ponteiro para void (void *)

- ii. Exemplo de função que não retorna nenhum valor:

```
void mostrarMensagem(){  
    printf("Esta mensagem é impressa pela função!  
    Feito isso, se encerra, não retornando nenhum dado.\n");  
}
```

8. Tem como criar saídas diferentes de um terminal do Windows?

a. Resposta:

- i. Sim, você pode por exemplo, mudar a cor do conteúdo que aparece no terminal utilizando a biblioteca *#windows.h*, em C.
- ii. Também é possível ter outros tipos de saídas de um terminal, depende da linguagem. Em Java (Swing, JavaFX), por exemplo, a saída pode ser um pop-up, rico em interface de usuário.

9. No MAPA, temos que descrever somente a correção de cada item que foi solicitado ou precisa corrigir a linha errada e reescrevê-la?

a. Resposta

- i. Não precisa enviar o código corrigido, apenas a descrição da correção de cada um dos itens mencionados.