





## **LISTA DE EXERCÍCIOS**

### *- Introdução aos Conceitos da Linguagem C -*

1. Qual a origem da Linguagem C, sua importância histórica para a computação e quais outras linguagens foram desenvolvidas a partir dela? Qual é a versão mais recente da linguagem C?
2. Em que ambientes e aplicações a linguagem C continua muito presente no mercado?
3. Porque C é muito relevante do ponto de vista didático?
4. O que são linguagens de baixo nível, “*médio nível*” e linguagens de alto nível? Como a linguagem C se caracteriza neste aspecto, e por que?
5. O que significa a expressão “*case-sensitive*” e como isso se aplica à Linguagem C?
6. O que são e para que servem bibliotecas (libs) de funções, e como elas se aplicam e são utilizadas na Linguagem C?
7. Pesquise a diferença entre linguagem compilada e linguagem interpretada. Vantagens e desvantagens. A linguagem C se enquadra em que modelo?
8. Quais são as etapas do ciclo de desenvolvimento de um programa na linguagem C.
9. O que significa a sigla IDE, e porque esse tipo de aplicação é útil?
10. O que é variável? Qual sua finalidade? E onde/como são armazenadas?
11. Pesquise e diferencie os conceitos de: linguagens fracamente tipadas, linguagens fortemente tipadas, linguagens estaticamente tipadas e linguagens dinamicamente tipadas. A linguagem C enquadra-se em quais modelos?
12. Na linguagem C, como se declaram variáveis para armazenar dados como: um valor inteiro, um símbolo/letra, e um valor fracionado?
13. Para que servem as expressões **long**, **short**, **double** e **unsigned** na Linguagem C, e em que casos são úteis e aplicáveis?
14. Qual o impacto do espaço ocupado (em bytes) por cada tipo de dado na linguagem C? O que é o conceito de *overflow* e porque é um grave problema.
15. Explique para o que serve, e como funciona a função **printf()**. Quais os principais caracteres de escape e para que servem?
16. O que indentação de código, qual a importância e forma correta de se fazê-lo?
17. O que são os comentários em código-fonte? Quais as vantagens e aplicações deste recurso? Como formas de realizar comentários na linguagem C?

18. Quais são os principais operadores aritméticos na Linguagem C? E o que significa precedência de operadores?
19. Explique para o que serve, e como funciona a função **scanf()**
20. Explique como funciona a estrutura condicional **if**
21. Quais são os principais operadores lógico-relacionais na Linguagem C?
22. Explique como funciona a estrutura condicional **switch-case**? O que faz, e por que usar a instrução **break** na estrutura **switch-case**?
23. Quais são os operadores de incremento, decremento e acumulação na Linguagem C?
24. Explique a sintaxe e funcionamento da estrutura de repetição **for**
25. Explique a sintaxe e funcionamento da estrutura de repetição **while**
26. Explique a sintaxe e funcionamento da estrutura de repetição **do-while**
27. Explique, em termos gerais, quando cada estrutura de repetição (**for**, **while**, **do-while**) é a mais adequada para uso.
28. Explique o comportamento da instrução **break** dentro de um laço de repetição.
29. Explique o comportamento da instrução **continue** dentro de um laço de repetição.
30. O que é *loop infinito*, quando essa condição é um problema e quando é parte de uma solução?

## PRIMEIRAS APLICAÇÕES NA LINGUAGEM

1. Somente **após** responder todas as questões teóricas acima, prossiga para os problemas práticos. Mas antes disso, prepare o **ambiente de desenvolvimento para a disciplina** 
  - Instale e use **Linux** - afinal, você está estudando para se tornar um **PROFISSIONAL** da computação  Distribuição recomendada: **Linux Mint Cinnamon Edition**.
    - Link: <https://www.linuxmint.com/download.php>
  - Após instalado, basta abrir o terminal (CTRL+ALT+T) e digitar os comandos em sequência...

```
sudo apt update  
sudo apt install build-essential gcc g++ geany
```
  - **Pronto!** Você já pode executar o **geany** -- que é a IDE que iremos utilizar na disciplina -- e já pode começar a *codar* em C.
2. Acesse o link abaixo e pesquise como resolver os problemas listados (pode inclusive usar GPT, solicitando para resolver os problemas na **Linguagem C** e utilizando **apenas 1 função**).
3. **ESTUDE O CÓDIGO** das soluções, **INTERPRETE-O**, e tente **REPLICAR a solução sem observar o original**. Não siga para o próximo problema sem ter conseguido realizar o anterior por si só.

[>> LINK <<](#)