

**APE - Algoritmos e Programação Estruturada**  
**Exercícios sobre Introdução à Lógica de Programação**  
**Professor Sandro Teixeira Carvalho**  
**Prática para Laboratório de Computação**

Crie cada um dos programas da lista abaixo no IDE **Code::Blocks** ou no IDE **Dev-C++**. Crie uma pasta com o seu nome completo dentro da pasta Documentos do computador do laboratório e salve seus programas nessa pasta. Em seguida, compacte essa pasta em formato zip ou rar e envie a pasta compactada para o professor. **Não envie arquivos com extensão .exe**. Lembre-se de excluir os arquivos com extensão .exe antes de compactar a pasta para enviar para o professor pela plataforma vigente da sua universidade. Envie somente os arquivos com extensão .c para o professor. Como sugestão, para o exercício 01 abaixo, crie um programa com o nome programa01.c. Siga a mesma nomenclatura para os outros exercícios.

Dicas rápidas: a estrutura for é mais indicada quando sabe-se previamente a quantidade de iterações. Já a estrutura while é mais indicada quando não se sabe ao certo a quantidade de iterações. A estrutura do..while é indicada, por exemplo, para fazer validações de entradas do usuário. São muitas as possibilidades e com a sua experiência acumulada ao longo do tempo você vai conseguir tomar a melhor decisão sobre a estrutura a ser utilizada para cada caso.

**Bom trabalho!**

- 1) Escreva um programa em linguagem C que imprima todos os números inteiros desde 1 até x utilizando a estrutura while, sendo x um número inteiro positivo informado pelo usuário.
- 2) Escreva um programa em linguagem C que imprima todos os números inteiros desde 1 até x utilizando a estrutura do-while, sendo x um número inteiro positivo informado pelo usuário.
- 3) Escreva um programa em linguagem C que imprima todos os números inteiros desde 1 até x utilizando a estrutura for, sendo x um número inteiro positivo informado pelo usuário.
- 4) Escreva um programa em linguagem C que imprima todos os números inteiros desde 1 até 500 mil. A impressão dos números poderá ser encerrada pelo usuário somente se ele pressionar a tecla ESC. A mensagem “Cancelado pelo usuário” ou a mensagem “Processo concluído” deve ser mostrada.
- 5) Escreva um programa em linguagem C que leia dois números inteiros de 1 até 50. Caso cada número informado não esteja no intervalo solicitado a mensagem “Número inválido” deverá ser apresentada ao usuário e o número deverá ser lido novamente. Cada número deve ser lido novamente sempre que seja inválido. O primeiro número deve ser inferior ao segundo número. Assim que ambos os números estiverem válidos o programa deverá calcular e imprimir o somatório de todos os números inteiros contidos entre os dois números informados. Utilize a estrutura do..while para a validação dos números informados.

- 6) Escreva um programa em linguagem C que leia um número de 10 até 100 em uma variável chamada `intervaloChute`. Em seguida, o programa deverá ler a quantidade de números a serem gerados aleatoriamente. O programa deverá gerar a quantidade de números aleatórios informada e os números aleatórios devem ser gerados dentro do intervalo informado.
- 7) Escreva um programa em linguagem C que leia um número inteiro de 1 até 5. Caso o usuário informe um número fora desse intervalo apresente a mensagem “Número inválido. Redigite”. Se o número informado estiver inválido o programa deverá ler o número novamente. Assim que o número informado estiver correto, o programa deverá apenas imprimir a mensagem “Número informado está correto”.
- 8) Escreva um programa em linguagem C que calcule o somatório de todos os números inteiros contidos desde `x` até `y`, sendo `x` e `y` informados pelo usuário.
- 9) Escreva um programa em linguagem C que calcule o somatório de todos os números inteiros pares e ímpares contidos desde `x` até `y`, sendo `x` e `y` informados pelo usuário.
- 10) Escreva um programa em linguagem C que leia uma frase qualquer e imprima essa frase/string na vertical na tela, isto é, cada caractere em uma nova linha da tela. Dica: para acessar cada caractere da string utilize a sintaxe: `variável[posição]`.
- 11) Escreva um programa em linguagem C que leia uma frase qualquer e conte o número de vogais que essa frase possui. Dica: para acessar cada caractere da string utilize a sintaxe: `variável[posição]`.
- 12) Escreva um programa em linguagem C que leia as duas notas de um aluno, calcule e imprima sua média aritmética simples. O programa deve imprimir também a mensagem “Aprovado” ou “Reprovado”. A média mínima para aprovação é 6,0. As notas lidas devem ser validadas (nota válida: 0,0 até 10,0). A mensagem “Nota inválida! Redigite.” deve ser apresentada e a nota deve ser lida novamente.

- 13) Escreva um algoritmo que calcule o que deve ser pago por um produto, considerando o preço normal de etiqueta e a escolha da condição de pagamento. Utilize os códigos da tabela a seguir para ler qual a condição de pagamento escolhida e efetuar o cálculo adequado. O programa deve permitir a leitura do preço e da condição de pagamento várias vezes e deve interromper a sua execução somente quando o usuário digitar um preço igual a zero. O preço e a condição de pagamento devem ser validados.

<b>Código</b>	<b>Condição de Pagamento</b>
1	À vista em dinheiro, recebe 10% de desconto
2	À vista no cartão de crédito, recebe 5% de desconto
3	Em duas vezes no cartão de crédito, preço normal de etiqueta sem juros
4	Em três vezes no cartão de crédito, preço normal de etiqueta mais juros de 10%

- 14) Escreva um programa em linguagem C que, dada a idade de um nadador, classifique-o em uma das categorias abaixo. O programa deve permitir a leitura da idade várias vezes e deve interromper a sua execução somente quando o usuário digitar uma idade igual a zero. A idade deve ser validada (idade válida de 1 até 100).

<b>Idade</b>	<b>Categoria</b>
5 até 7 anos	Infantil A
8 até 10 anos	Infantil B
11 até 13 anos	Juvenil A
14 até 17 anos	Juvenil B
Maiores de 18 anos	Adulto