

Lista de Exercícios III

1. Escreva um programa em C para ler uma temperatura em graus Fahrenheit, calcular e escrever o valor correspondente em graus Celsius (de acordo com a fórmula abaixo).

$$\frac{C}{5} = \frac{F - 32}{9}$$

2. Escreva um programa em C para ler uma temperatura em graus Celsius, calcular e escrever o valor correspondente em graus Fahrenheit.
3. Escreva um programa em C para ler o número de lados de um polígono regular, e a medida do lado. Calcular e imprimir o seguinte:

Se o número de lados for igual a 3 escrever TRIÂNGULO e o valor do seu perímetro. Se o número de lados for igual a 4 escrever QUADRADO e o valor da sua área. Se o número de lados for igual a 5 escrever PENTÁGONO.

Em qualquer outra situação escrever Polígono não identificado.

4. Escreva um programa em C que leia as medidas dos lados de um triângulo e escreva se ele é EQUILÁTERO, ISÓSCELES ou ESCALENO.
5. Escreva um programa em C que leia a idade de 2 homens e 2 mulheres (considere que a idade dos homens será sempre diferente, assim como das mulheres). Calcule e escreva a soma das idades do homem mais velho com a mulher mais nova, e o produto das idades do homem mais novo com a mulher mais velha.
6. Ler 2 valores inteiros, calcular e escrever a soma dos inteiros existentes entre os 2 valores lidos (incluindo os valores lidos na soma). Considere que o primeiro pode ser menor que o segundo e vice-versa.
7. Escreva um programa em C para validar um lote de cheques. O programa deverá inicialmente solicitar a soma do lote e o número de cheques. A seguir deverá ler o valor de cada cheque calculando a soma total. Após a digitação de todos os cheques o programa deverá imprimir as seguintes mensagens:

LOTE Ok: soma informada=igual
soma calculada. Diferença negativa
se a soma calculada for menor que a
informada. Diferença positiva se a
soma calculada for maior que a
informada. Observação: O valor da
diferença deve ser impresso (caso
exista).

8. Escreva um programa em C para ler o saldo inicial de uma conta bancária. A seguir ler um número indeterminado de pares de valores indicando respectivamente o tipo da operação (codificado da seguinte forma: 1.Depósito 2.Retirada 3.Fim) e o valor. Quando for informado para o tipo o código 3, o programa deve ser encerrado e impresso o saldo final da conta com as seguintes mensagens: CONTA ZERADA, CONTA ESTOURADA(se o saldo for negativo) ou CONTA PREFERENCIAL (se o saldo for positivo).

Vectores:

1. Escreva um programa em C para ler um vetor **X** de 10 elementos inteiros. Logo após copie os elementos do vetor **X** para um vetor **Y** fazendo com que o 1º. elemento de **X** seja copiado para o 10º. de **Y**, o 2º. de **X** para o 9º. de **Y** e assim sucessivamente. Após o término da cópia, imprimir o vetor **Y**.
2. Escreva um programa em C para ler um vetor **A** de 10 elementos inteiros e um valor **X**. A seguir imprimir os índices do vetor **A** em que aparece um valor igual a **X**.
3. Escreva um programa em C para ler um vetor **A** de 10 elementos e um valor **X**. Copie para um vetor **S** (sem deixar elementos vazios entre os valores copiados) os elementos de **A** que são maiores que **X**. Logo após imprimir o vetor **S**.
4. Escreva um programa em C para ler um vetor de 10 elementos inteiros. Excluir o 1º. elemento do vetor deslocando os elementos subsequentes de uma posição para o início. Imprimir o vetor após a retirada do primeiro elemento.

5. Escreva um programa em C para ler um vetor **X** de 10 elementos e um valor **P** (aceitar apenas valores entre 0 e 9) que representa a posição de um elemento dentro do vetor **X**. Imprimir o valor do elemento que ocupa a posição informada. Logo após excluir esse elemento do vetor fazendo com que os elementos subsequentes (se houverem) sejam deslocados de 1 posição para o início. Imprimir o vetor **X** após a exclusão ter sido executada.
6. Escreva um programa em C para ler um vetor **R** (de 5 elementos) e um vetor **S** (de 10 elementos). Gere um vetor **X** que possua os elementos comuns a **R** e a **S**. Considere que pode existir repetição de elementos no mesmo vetor. Nesta situação somente uma ocorrência do elemento comum aos dois deve ser copiada para o vetor **X**. Após o término da cópia, escrever o vetor **X**.

Funções:

1. Escreva um programa que leia um número indeterminado de valores que representam raios de círculos. Para cada valor informado imprimir a área e seu perímetro. O programa será encerrado ao ser fornecido para o raio um valor negativo. Para obter o valor da área do círculo o programa deverá chamar a função **AreaCirculo**. Para obter o valor do seu perímetro o programa deverá chamar a função **CompCircunferencia**

Nome: **AreaCirculo**

Descrição: Retorna a área do círculo.

Entrada: float Raio

Saída: (float) A área do círculo

Nome: **CompCircunferencia**

Descrição: Retorna o comprimento da circunferência (perímetro).

Entrada: float Raio

Saída: (float) O comprimento da circunferência

2. Escreva um programa em C para ler 5 pares de valores (considere que serão informados apenas valores positivos). Para cada par lido deve ser impresso o valor do maior elemento do par ou a frase "Eles são iguais" se os valores do par forem iguais. Para obter o maior elemento do par utilize a função **MaiorNumero**.

Nome: **MaiorNumero**

Descrição: Retorna o maior elemento entre 2 valores positivos. Se eles forem iguais deve ser retornado o valor -1.

Entrada: (int) Dois valores positivos.

Saída: (int) O maior deles ou -1 se eles forem iguais.

Observação: Considere que os valores de entrada são sempre positivos.

3. Escreva um programa para ler 5 números inteiros positivos (utilize a função **LePositivo**). Para cada valor lido escrever a soma dos inteiros de 1 ao número informado. O resultado do cálculo desse somatório deve ser obtido através da função **Somatorio**.

Nome: **LePositivo**

Descrição: Faz a leitura de um valor. Se ele for negativo ou zero, a leitura deve ser repetida até que o valor lido seja positivo.

Entrada: Nenhuma.

Saída: (int) o valor lido.

Nome: **Somatório**

Descrição: Calcula o somatório dos inteiros de 1 ao número fornecido como entrada.

Entrada: (int) Número limite do somatório.

Saída: (int) O valor do somatório.

4. Escreva um programa que leia 5 números inteiros positivos (utilizar **LePositivo**). Para cada número informado escrever a soma de seus divisores (exceto ele mesmo). Utilize a função **SomaDivisores** para obter a soma.

Nome: **SomaDivisores**

Descrição: Calcula a soma dos divisores do número informado (exceto ele mesmo).

Entrada: Um número inteiro e positivo.

Saída: A soma dos divisores.

Exemplo: Para o valor 8: $1+2+4 = 7$

5. Escreva um programa que imprima na tela os números primos existentes entre 1 e 100. Para verificar se um número é primo utilize a função **EhPrimo**.

Nome: **EhPrimo**

Descrição: Verifica se um número é o ou não primo.

Entrada: (int) um número inteiro.

Saída: (int) 1 se o número de entrada for primo e 0 caso contrário.

6. Escreva um programa que leia 5 pares de valores positivos (**LePositivo**). Imprima se os elementos de cada par são números amigos (ou não). Dois números A e B são amigos se a soma dos divisores de A excluindo A é igual a B e a soma dos divisores de B excluindo B é igual a A. Para a verificar se dois números são amigos utilize a função **SaoAmigos**.

Nome: SaoAmigos

Descrição: retorna 1 se os 2 números de entrada forem amigos, 0 caso contrário.

Entrada: (int) Dois números inteiros positivos.

Saída: (int) 1 se os dois números são amigos, 0 caso contrário.

Observação: Utilize a função SomaDivisores do exercício anterior.

Exemplo:

220 e 284 são amigos, pois

220: $1+2+4+5+10+11+20+22+44+55+110=284$

284: $1+2+4+71+142=220$

1184 e 1210 também são amigos

7. Escreva um programa que leia as medidas dos lados de 5 triângulos. Para cada triângulo imprimir a sua classificação (Não é triângulo, Triângulo Equilátero, Isósceles ou Escaleno). O programa deve aceitar apenas valores positivos para as medidas dos lados (utilizar **LePositivo**). Para obter o código da classificação utilize a função **TipoTriangulo**.

Nome: **TipoTriangulo**

Descrição: A partir das medidas dos lados de um triângulo, verifica o tipo do triângulo.

Entrada: (int) 3 valores

Saída: (int) 0 se não formam um triângulo.

1 se for um triângulo equilátero.

2 se for um triângulo isósceles.

3 se for um triângulo escaleno.

Para verificar se as medidas formam um triângulo

chamar a função **EhTriangulo**. Nome: **EhTriangulo**

Descrição: Verifica se as 3 medidas informadas permitem formar um triângulo. Para formar um triângulo é necessário que a medida de cada lado seja menor que a soma dos outros 2.

Entrada: (int) 3 valores.

Saída: (int) 1 se os 3 valores formarem um triângulo e 0 caso contrário.