

Lista para prova

7. Defina uma estrutura *empregado* para armazenar nome, data de nascimento, idade e salário de um empregado de uma empresa. Leia informações sobre 10 empregados da empresa em um vetor e armazene as informações sobre cada um em um arquivo chamado *RH.txt*, um empregado por linha.
8. Faça um programa que leia um arquivo fornecido pelo usuário e salve em outro arquivo o conteúdo do primeiro, substituindo as vogais por '\*' (asterisco). O novo arquivo deverá se chamar *sem\_vogal.txt*.
9. Uma prova constou de 20 questões do tipo múltipla escolha de 5 alternativas (**a, b, c, d, e**). Vários alunos fizeram a prova. As notas das provas variam entre 0 e 10. A nota mínima para aprovação é 7.

Escrever um programa para:

- a) ler e imprimir o gabarito da prova;
- b) ler o número de alunos que fizeram a prova;
- c) alocar dinamicamente um vetor do tipo estruturado **aluno** de tamanho igual ao número de alunos lido no item anterior;
- d) ler o nome (máximo de 20 caracteres) e as respostas de cada aluno;
- e) calcular e imprimir uma tabela dos resultados da prova, na qual cada linha deve conter o nome, as respostas e a nota final de um determinado aluno. Para cada aluno, deve ainda haver uma informação dizendo se ele está aprovado ou reprovado.

Utilizar na programação uma estrutura **aluno** contendo os seguintes campos: nome, respostas, nota final e indicação de aprovado ou reprovado.

4. Um **racional** é qualquer número da forma  $p/q$ , sendo **p** inteiro e **q** inteiro não-nulo. Assim, é conveniente representar cada racional por uma estrutura:

```
struct racional {
    int p;
    int q;
};
```

Usando a esta estrutura, escreva funções que:

- receba dois inteiros  $x$  e  $y$  e devolva o racional que representa  $x/y$ ;
- receba dois racionais  $x$  e  $y$  e devolva o racional que representa a soma de  $x$  e  $y$ ;
- receba dois racionais  $x$  e  $y$  e devolva o racional que representa o produto de  $x$  por  $y$ ;
- receba dois racionais  $x$  e  $y$  e devolva o racional que representa o quociente de  $x$  por  $y$ ;

A função principal deve ler quatro números inteiros e, a partir deles, obter dois números racionais. Em seguida deve ser impresso o resultado das operações de soma, multiplicação e divisão desses dois números racionais.

5. Considere a seguinte estrutura, representando uma data (dia, mês e ano):

```
struct dma {
    int d;
    int m;
```

```

    int a;
};

```

Faça uma função chamada *periodo* que receba dois ponteiros para estruturas do tipo *dma*, cada um representando uma data válida, e retorne o número de dias entre as duas datas. Considere que a segunda data será sempre maior que a primeira e que todos os meses têm 30 dias. A função principal deve alocar memória para os dois ponteiros para estrutura, ler os campos (dia, mês e ano) de cada um e imprimir o retorno da função *periodo*.

6. Escreva um programa que calcule uma aproximação da raiz quadrada de um número usando o método de Newton. Pelo método de Newton, para calcular a raiz R de um número X faça:

$$R = R - (R^2 - X)/(2 \cdot R)$$

Quanto mais interações você fizer, melhor será a aproximação. Como sugestão, utilize  $X/2$  como valor inicial de R.

7. Escreva um programa que lê 2 pontos (suas coordenadas) e calcula a distância entre esses 2 pontos, usando uma função para efetuar o cálculo. O ponto deve ser definido como uma estrutura e o resultado deve ser armazenado em arquivo chamado *pontos.txt* no seguinte formato:

x1, y1 – x2, y2 – distância

Exemplo:

1.4, 3.5 – 4.4, 7.5 – 5.0

8. A obesidade não é mais apenas um problema estético, o excesso de peso pode provocar o surgimento de vários problemas de saúde como diabetes, problemas cardíacos e a má formação do esqueleto.

Item (a):

O índice de massa corporal (IMC) é uma medida internacional usada para calcular obesidade, adotada pela Organização Mundial de Saúde (OMS). O IMC é calculado pela seguinte fórmula:

$$\text{IMC} = \text{peso}/(\text{altura} \cdot \text{altura})$$

Onde a massa está em quilogramas e a altura está em metros. Faça uma função que calcula o IMC e que retorna o grau de obesidade do indivíduo segundo a tabela abaixo:

IMC	Grau
< 18,5	-1 (abaixo do peso)
18,5 - 29,9	0 (normal)
30,0 - 34,9	1 (obesidade grau I)
35,0 – 39,9	2 (obesidade grau II)
> 40,0	3 (obesidade grau III)

Use o protótipo abaixo:

```
int GrauObesidade(float massa, float altura);
```

Item (b):

Em uma pesquisa foram coletados os seguintes dados de uma população: Sexo (ex: M ou F), idade (ex: 25 anos), altura (ex: 1.7 m) e peso (ex: 75.5 Kg). Dado um arquivo texto "pesquisa.txt" que possui em sua primeira linha um inteiro que indica a quantidade de pessoas entrevistadas, e nas demais linhas os dados da pesquisa (um entrevistado por linha), conforme o exemplo:

```
3
M 25 1.70 75.5
F 8 1.50 50.4
M 15 1.63 62.7
```

Escreva um programa que:

- Defina uma estrutura para armazenar os dados de cada entrevistado.
- Aloque dinamicamente um vetor de estruturas e preencha-o com os valores lidos do arquivo "pesquisa.txt".
- Calcule o número de crianças obesas (grau maior ou igual a 1). Considere como crianças as pessoas com idade menor que 13 anos. Dica: utilize a função do item (a).
- Crie um arquivo texto "homens.txt" que contém somente os dados das pessoas do sexo masculino entrevistadas. Obs: A primeira linha do arquivo deve conter o número de registros e as demais os dados (um registro por linha).
- Desaloque o vetor e feche os arquivos.

## Ponteiros

1. Considere o seguinte código em C:

```
#include <iostream>
using namespace std;

int funcao_qualquer(int inteiro) {
    inteiro = inteiro + 2;
    return inteiro;
}

int main() {
    int variavel_a = 5;
    int variavel_b = funcao_qualquer(variavel_a);
    cout << variavel_a << " - " << variavel_b << endl;
    return 0;
}
```

Quais serão os valores impressos por este programa?

2. Considere o seguinte código em C:

```
#include <iostream>
using namespace std;

int funcao_qualquer_modificada(int* inteiro) {
    *inteiro = *inteiro + 2;
    return *inteiro;
}
```

```

}

int main() {
    int variavel_a = 5;
    int variavel_b = funcao_qualquer_modificada(&variavel_a);
    cout << variavel_a << " - " variavel_b << endl;
    return 0;
}

```

Quais serão os valores impresso por este programa?

3. Mostre o valor especificado em cada item após a execução das instruções abaixo, supondo que o endereço da variável x, e 1000 e da variável y, e 1004.

```

int x, y;
int *p1;
int *p2;
x = 10;
y = 20;
p1 = &x;
p2 = &y;
(*p1)++;

```

- a) x
- b) y
- c) &x
- d) &y
- e) p1
- f) p2
- g) \*p1 + \*p2
- h) \*(&x)
- i) &(\*p2)

4. Mostre a saída do programa a seguir:

```

#include <iostream>
using namespace std;

int main() {
    int i1, i2, *p1, *p2;
    i1 = 5;
    p1 = &i1;
    i2 = *p1 / 2 + 10;
    cout << "i1 = " << i1 << ", i2 = " << i2 << ", *p1 = " <<
    *p1;
    cout << ", *p2 = " << *p2 << endl;
    return 0;
}

```

5. Considere o seguinte código. Há um erro na função misterio. Qual é? Dê uma solução para o erro.

```
#include <iostream>
using namespace std;

void misterio (int* p, int* q) {
    int* temp;
    *temp = *p;
    *p = *q;
    *q = *temp;
}

int main() {
    int i = 6, j= 10;
    misterio(&i, &j);
    cout << i << " " << j);
    return 0;
}
```

6. Suponha que  $v$  é um vetor. Descreva a diferença conceitual entre as expressões  $v[3]$  e  $v + 3$ .

7. Suponha que os elementos do vetor  $v$  são do tipo *int* e cada *int* ocupa 8 bytes no seu computador. Se o endereço de  $v[0]$  é 55000, qual o valor da expressão  $v + 3$ ?