# Lista de Exercícios

**Aluno:** Luiz Henrique de Luccas

**Estrutura Sequencial**

1) Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês.

```
#include <iostream>


using namespace std;

int main(){
   float valor, horas, salario;
   char resp='s';
   do{
      cout<<"Qual o valor em horas: ";
      cin >>valor;
      cout<<"Qual o numero de horas trabalhadas no mes: ";
      cin >>horas;


      salario = horas * valor;


      cout<<"Seu salario foi de: " << salario << endl;
      cout<<"Deseja Continuar? [S/N]: ";
      cin >>resp;
   }while(resp == 's' || resp == 'S');
   return 0;
}
```

2) Faça um Programa que peça 2 números inteiros e um número real. Calcule e mostre: o produto do dobro do primeiro com metade do segundo. a soma do triplo do primeiro

com o terceiro. o terceiro elevado ao cubo.

```
#include <iostream>

using namespace std;
int main(){
   int n1, n2;
   float n3, calculo1, calculo2, calculo3;
   char resp='s';
   do{
      cout<<"N1 inteiro: ";
      cin >>n1;
      cout<<"N2 inteiro: ";
      cin >>n2;
      cout<<"N3 Real: ";
      cin >>n3;

      calculo1 = (n1 * 2) + (n2 / 2);
      calculo2 = (n1 * 3) + n3;
      calculo3 = n3* n3 * n3;


      cout<<"O produto do dobro de "<<n1<<" com metade de "<<n2<<" e: " <<
calculo1 << endl;
      cout<<"A soma do triplo de "<<n1<<" com "<<n3<<" é: " << calculo2 << endl;
      cout<< n3<<" elevado ao cubo é: " << calculo3 << endl;

      cout<<"Deseja Continuar? [S/N]: ";
      cin >>resp;
   }while(resp == 's' || resp == 'S');
   return 0;
}
```

3) Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês.

Calcule e mostre o total do seu salário no referido mês, sabendo-se que são descontados 11% para o Imposto de Renda, 8% para o INSS e 5% para o sindicato, faça um programa que nos dê: salário bruto. quanto pagou ao INSS. quanto pagou ao sindicato. o salário líquido. calcule os descontos e o salário líquido, conforme a tabela abaixo:

+ Salário Bruto : R$
- IR (11%) : R$
- INSS (8%) : R$
- Sindicato ( 5%) : R$
= Salário Liquido : R$
Obs.: Salário Bruto - Descontos = Salário Líquido.

```
#include <iostream>

using namespace std;
int main(){
   float valor, horas, salario, salariobruto, descontos, sindicato, inss;
   char resp='s';

   do{
      cout<<"Qual o valor em horas: ";
      cin >>valor;
      cout<<"Qual o numero de horas trabalhadas no mes: ";
      cin >>horas;

      salariobruto = horas * valor;
      sindicato = salariobruto * 0.08;
      inss = salariobruto * 0.05;
      descontos = (salariobruto * 0.11) + (salariobruto * 0.08) + (salariobruto * 0.05);
      salario = salariobruto - descontos;

      cout<<"Salario bruto R$"<< salariobruto << endl;
      cout<<"INSS R$"<< inss << endl;
      cout<<"Sindicato R$"<< sindicato << endl;
      cout<<"Salario liquido R$"<< salario << endl;


      cout<<"Deseja Continuar? [S/N]: ";
      cin >>resp;

   }while(resp == 's' || resp == 'S');
   return 0;
}
```

4) Faça um programa para uma loja de tintas. O programa deverá pedir o tamanho em metros quadrados da área a ser pintada.

   Considere que a cobertura da tinta é de 1 litro para cada 3 metros quadrados e que a tinta é vendida em latas de 18 litros, que custam R$ 80,00.
   Informe ao usuário a quantidades de latas de tinta a serem compradas e o preço total.

```
#include <iostream>

using namespace std;

int main() {
   float area, preco = 80.00, valor, quantidad;
   int tinta = 18, quantidade;


   cout << "Qual a area em M2 a ser pintada: ";
   cin >> area;

   valor = area * tinta;
   quantidad = valor / 54;

   if (quantidad == static_cast<int>(quantidad)) {
      quantidade = quantidad / 18;
   }
else {
      quantidade = static_cast<int>(quantidad / 18) + 1;
   }

   preco = preco * quantidade;

   cout << "\n Quantidade de latas de tintas: " << quantidade;
   cout << "\n Valor a ser gasto: " << preco;


   return 0;
}
```

5) Faça um programa que peça o tamanho de um arquivo para download (em MB) e a velocidade de um link de Internet (em Mbps), calcule e informe o tempo aproximado de download do arquivo usando este link (em minutos).

```
#include <iostream>

using namespace std;
int main() {
  float tam_arq_mb, vel_int_mbps, vel_mb_por_segundo, tempo_download_seg, tempo_download_min;

  cout << "Digite o tamanho do arquivo para download (em MB): ";
  cin >> tam_arq_mb;

  cout << "Digite a velocidade da sua conexão de Internet (em Mbps): ";
  cin >> vel_int_mbps;

vel_mb_por_segundo = vel_int_mbps / 8;

  tempo_download_seg = tam_arq_mb / vel_mb_por_segundo;

  tempo_download_min = tempo_download_seg / 60;

  cout << "O tempo aproximado de download do arquivo e de " << tempo_download_min << " minutos." << endl;

  return 0;
}
```

6) Criar um programa que a partir da idade e peso do paciente calcule a dosagem de determinado medicamento e imprima a receita informando quantas gotas do medicamento o paciente deve tomar por dose. Considere que o medicamento em questão possui 500mg por ml, e que cada ml corresponde a 20 gotas.
   ▪ Adultos ou adolescentes desde 12 anos, inclusive, se tiverem peso igual ou acima e 60 quilos devem tomar 1000mg; com peso abaixo de 60 quilos devem tomar 875mg.
   ▪ Para crianças e adolescentes abaixo de 12 anos a dosagem é calculada pelo peso corpóreo conforme a tabela a seguir:

| Peso | Dosagem |
|---|---|
| Até 15kg | 200mg |
| Até 30kg | 500mg |
| Até 50kg | 750mg |

```
#include <iostream>

using namespace std;
int main() {
float peso;
int idade, gotas, c;

cout << "Informe a idade: ";
cin >> idade;

cout << "Informe o peso: ";
cin >> peso;

if(idade >= 12 & peso >= 60) {
gotas = 40;
}
else if(idade >= 12 & peso < 60) {
gotas = 35;
}
else if (idade < 12 & peso >= 31 & peso < 50) {
gotas = 30;
}
else if (idade < 12 & peso >= 16 & peso <= 30) {
gotas = 20;
}
else if (idade < 12 & peso > 0 & peso <= 15) {
gotas = 8;
}

cout << "Voce deve tomar " << gotas << " gotas por dose" << endl;
return 0;
}
```

**Estrutura de Decisão**

7) Faça um Programa que leia três números e mostre-os em ordem decrescente.

```
#include <iostream>
using namespace std;
int main() {
float n1, n2, n3;
cout << "N1: ";
cin >> n1;

cout << "N2: ";
cin >> n2;

cout << "N3: ";
cin >> n3;

if (n1 > n2 & n1 > n3 & n2 > n3){
cout << n1 << n2 << n3;
}
else if (n1 > n2 & n1 > n3 & n3 > n2){
cout << n1 << n3 << n2;
}
else if (n2 > n1 & n2 > n3 & n1 > n3){
cout << n2 << n1 << n3;
}
else if(n2 > n1 & n2 > n3 & n3 > n1){
cout << n2 << n3 << n1;
}
else if(n3 > n1 & n3 > n2 & n1 > n2){
cout << n3 << n1 << n2;
}
else if (n3 > n1 & n3 > n2 & n2 > n1) {
cout << n3 << n1 << n2;
}
```

```
return 0;

}
```

8) Faça um Programa que peça os 3 lados de um triângulo. O programa deverá informar se os valores podem ser um triângulo. Indique, caso os lados formem um triângulo, se o mesmo é: equilátero, isósceles ou escaleno.

   Dicas: Três lados formam um triângulo quando a soma de quaisquer dois lados for maior que o terceiro; Triângulo Equilátero: três lados iguais; Triângulo Isósceles: quaisquer dois lados iguais; Triângulo Escaleno: três lados diferentes;

```
#include <iostream>

using namespace std;
int main() {
   int n[3], a, b, c;


cout << "Informe o lado A: ";
cin >> a;
   cout << "Informe o lado B: ";
cin >> b;
cout << "Informe o lado C: ";
cin >> c;

if (a+b>c & a+c>b & b+c>a){
if (a==b & b==c & a==c){
   cout << "Triangulo Equilatero";
}
else if (a==b || b==c || a==c){
   cout << "Triangulo Isósceles";
}
else if (a!=b & b!=c & a!=c){
   cout << "Triangulo Escaleno";
}
}
else {
   cout << "Nao e possivel formar um triangulo";
}

   return 0;
}
```

9) Faça um programa que calcule as raízes de uma equação do segundo grau, na forma $ax^2 + bx + c$.

O programa deverá pedir os valores de a, b e c e fazer as consistências, informando ao usuário nas seguintes situações:

Se o usuário informar o valor de A igual a zero, a equação não é do segundo grau e o programa não deve fazer pedir os demais valores, sendo encerrado;
Se o delta calculado for negativo, a equação não possui raízes reais. Informe ao usuário e encerre o programa;
Se o delta calculado for igual a zero a equação possui apenas uma raiz real; informe-a ao usuário;
Se o delta for positivo, a equação possui duas raízes reais; informe-as ao usuário;

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
   double a, b, c;

   cout << "Digite o valor de a: ";
   cin >> a;

   if (a == 0) {
      cout << "A equação não é do segundo grau. Encerrando o programa." << endl;
      return 0;
   }

   cout << "Digite o valor de b: ";
   cin >> b;
   cout << "Digite o valor de c: ";
   cin >> c;

   double delta = b * b - 4 * a * c;


   if (delta < 0) {
      cout << "A equação não possui raízes reais. Encerrando o programa." << endl;
   } else if (delta == 0) {

      double raiz = -b / (2 * a);
      cout << "A equação possui uma raiz real: " << raiz << endl;
   } else {

      double raiz1 = (-b + sqrt(delta)) / (2 * a);
      double raiz2 = (-b - sqrt(delta)) / (2 * a);
      cout << "A equação possui duas raízes reais: " << raiz1 << " e " << raiz2 << endl;
   }

   return 0;
}
```

10) Faça um Programa que peça um número correspondente a um determinado ano e em seguida informe se este ano é ou não bissexto.

```
#include <iostream>

using namespace std;

int main() {
 int ano;

 cout << "Digite um ano: ";
 cin >> ano;

 if ((ano % 4 == 0 && ano % 100 != 0) || (ano % 400 == 0)) {
        cout << ano << " é um ano bissexto." << endl;
 }
else {
        cout << ano << " não é um ano bissexto." << endl;
 }

  return 0;
}
```

11) Faça um Programa que leia um número inteiro maior que 0 e menor que 1000 e imprima a quantidade de centenas, dezenas e unidades do mesmo.
Observando os termos no plural a colocação do "e", da vírgula entre outros.

Exemplo: 326 = 3 centenas, 2 dezenas e 6 unidades 12 = 1 dezena e 2 unidades

```
#include <iostream>

using namespace std;
int main() {
  int numero;

  cout << "Digite um número inteiro maior que 0 e menor que 1000: ";
  cin >> numero;
  if (numero <= 0 || numero >= 1000) {
    cout << "Número fora do intervalo permitido." << endl;
    return 0;
  }

  int centenas = numero / 100;
  int resto = numero % 100;
  int dezenas = resto / 10;
  int unidades = resto % 10;

  if (centenas > 0) {
    cout << centenas;
    if (centenas == 1) {
      cout << " centena";
    } else {
      cout << " centenas";
    }
    if (dezenas > 0 || unidades > 0) {
      cout << ", ";
    }
  }

  if (dezenas > 0) {
    cout << dezenas;
    if (dezenas == 1) {
      cout << " dezena";
    } else {
      cout << " dezenas";
    }
    if (unidades > 0) {
      cout << " e ";
    }
  }

  if (unidades > 0 || (centenas == 0 && dezenas == 0)) {
    cout << unidades;
    if (unidades == 1) {
```

```
            cout << " unidade";
          } else {
            cout << " unidades";
          }
        }

        cout << endl;

        return 0;
      }
```

12) Faça um Programa para um caixa eletrônico.
    O programa deverá perguntar ao usuário a valor do saque e depois informar quantas notas de cada valor serão fornecidas.

    As notas disponíveis serão as de 1, 5, 10, 50 e 100 reais. O valor mínimo é de 10 reais e o máximo de 600 reais.

    O programa não deve se preocupar com a quantidade de notas existentes na máquina.

    Exemplo 1: Para sacar a quantia de 256 reais, o programa fornece duas notas de 100, uma nota de 50, uma nota de 5 e uma nota de 1;

    Exemplo 2: Para sacar a quantia de 399 reais, o programa fornece três notas de 100, uma nota de 50, quatro notas de 10, uma nota de 5 e quatro notas de 1.

```
#include <iostream>

            using namespace std;
            int main() {
int aux, nota1, nota100, valor, nota5, nota50, nota10;

cout << "Valor do saque: ";
cin >> valor;

nota100 = valor / 100;

if (nota100 > 0){
cout << "Notas de 100: " <<nota100 << endl;
}

aux = nota100 * 100;
valor = valor - aux;
nota50 = valor / 50;

if (nota50 > 0){
cout << " Notas de 50: " <<nota50 << endl;
}

aux = nota50 * 50;
valor = valor - aux;
nota10 = valor / 10;

if (nota10 > 0){
cout << " Notas de 10: " <<nota10 << endl;
```

```
}

aux = nota10 * 10;
valor = valor - aux;
nota1 = valor;

if (nota1 > 0){
cout << " Notas de 1: " <<nota1;
}

return 0;
}
```

**Estrutura de Repetição**

13) Faça um Programa que verifique se um CPF fornecido pelo usuário é válido.

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

bool verificaCPF(string cpf) {
  if (cpf.length() != 11)
    return false;

  bool digitosIguais = true;
  for (int i = 1; i < 11; i++) {
    if (cpf[i] != cpf[0]) {
      digitosIguais = false;
      break;
    }
  }
  if (digitosIguais)
    return false;

  int soma = 0;
  for (int i = 0; i < 9; i++) {
    soma += (cpf[i] - '0') * (10 - i);
  }
  int digito1 = 11 - (soma % 11);
  if (digito1 > 9)
    digito1 = 0;

  if (digito1 != (cpf[9] - '0'))
    return false;
```

```
    soma = 0;
    for (int i = 0; i < 10; i++) {
        soma += (cpf[i] - '0') * (11 - i);
    }
    int digito2 = 11 - (soma % 11);
    if (digito2 > 9)
        digito2 = 0;

    if (digito2 != (cpf[10] - '0'))
        return false;

    return true;
}

int main() {
    string cpf;
    cout << "Digite o CPF (somente números): ";
    cin >> cpf;

    if (verificaCPF(cpf))
        cout << "CPF válido." << endl;
    else
        cout << "CPF inválido." << endl;

    return 0;
}
```

14) Faça o programa que apresenta a seguinte saída, perguntando ao usuário o número máximo (no exemplo, 9). Este número deve ser sempre ímpar.

```
1 2 3 4 5 6 7 8 9
 2 3 4 5 6 7 8
  3 4 5 6 7
   4 5 6
    5
```

```
#include <iostream>

using namespace std;
void imprimirPadrao(int numeroMaximo) {
   if (numeroMaximo % 2 == 0)
      numeroMaximo--;

   for (int i = 1; i <= numeroMaximo; i++) {
      for (int j = i; j <= numeroMaximo; j++) {
         cout << j << " ";
      }
      cout << endl;
   }
}

int main() {
   int numeroMaximo;
   cout << "Digite um número máximo (deve ser ímpar): ";
   cin >> numeroMaximo;

   imprimirPadrao(numeroMaximo);

   return 0;
}
```

15) A série de Fibonacci é uma sequência de termos que tem como os 2 primeiros termos, respectivamente, os números 0 e 1. A partir daí os demais termos são formados seguindo uma certa regra. A série de Fibonacci pode ser vista a seguir:

0 1 1 2 3 5 8 13 21...

```
#include <iostream>

using namespace std;
int main(){
   int n1=0, n2=1,aux, resp;

   cout << "Até qual repetição da sequência de Fibonacci será feito: ";
   cin >> resp;

   cout <<n1 << ". ";;
   cout <<n2 << ". ";;
   for (int c=0; c<= resp; c++){
      aux = n2;
      n2 = n1 + aux;
      n1 = aux;
      cout << n2 << ". ";
   }

   return 0;
}
```

16) Escrever um programa que calcule e apresente a somatória do número de grãos de trigo que se pode obter em um tabuleiro de xadrez, obedecendo a seguinte regra: colocar um grão de trigo no primeiro quadro e nos quadros seguintes o dobro do quadro anterior. Ou seja, no primeiro coloca-se um grão, no segundo quadro coloca-se dois grãos (neste momento tem-se três grãos), no terceiro coloca-se quatro grãos, repetir até atingir o sexagésimo quarto quadro. (Este exercício foi baseado em uma situação do capítulo 16 do livro "O Homem que calculava" de Malba Tahan.

```
#include <iostream>

using namespace std;
int main(){
   unsigned long long n1=0, n2=1,aux, calculo=1;

   cout <<"Graos no 1 quadro: "<<n2 << endl;

   for (int c=2; c<=64; c++){
      aux = n2;
      n1 = aux;
      n2 = n1 + aux;
      cout <<"Graos no "<<c<<" quadro: "<<n2 << endl;
      calculo = calculo + n2;
   }

   cout <<"Total de Graos: "<<calculo;

   return 0;
}
```

17) O Sr. Manoel Joaquim expandiu seus negócios para além dos negócios de 1,99 e agora possui uma loja de conveniências.
Faça um programa que implemente uma caixa registradora rudimentar.
O programa deverá receber um número desconhecido de valores referentes aos preços das mercadorias.

Um valor zero deve ser informado pelo operador para indicar o final da compra.

O programa deve então mostrar o total da compra e perguntar o valor em dinheiro que o cliente forneceu, para então calcular e mostrar o valor do troco.

Após esta operação, o programa deverá voltar ao ponto inicial, para registrar a próxima compra.

A saída deve ser conforme o exemplo abaixo:
Lojas Tabajara
Produto 1: R$ 2.20
Produto 2: R$ 5.80
Produto 3: R$ 0
Total: R$ 9.00
Dinheiro: R$ 20.00
Troco: R$ 11.00

```
#include <iostream>

using namespace std;
int main(){
   bool sai=false, comeco=true;

   do{

      float produto=0, total=0, pagamento=0, troco=0;

      int c1=0, c=1;

      cout <<"=============="<<endl;
      cout <<"Lojas Tabajara"<<endl;
      cout <<"=============="<<endl;
      do {
         cout <<"Informe o valor do produto "<<c<<" (0 para sair): R$";
         c = c + 1;
         cin >> produto;

         total = total + produto;
```

```
   c1 = c1 + 1;

   if (produto == 0 && c1 == 1) {
      c1 = 0;
   }
} while (produto != 0);


/*
A saída deve ser conforme o exemplo abaixo:

Lojas Tabajara Produto 1: R$ 2.20
Produto 2: R$ 5.80
Produto 3: R$ 0
Total: R$ 9.00
Dinheiro: R$ 20.00
Troco: R$ 11.00
*/

if(c1 > 0){
   cout <<"Total: R$"<<total<< endl;
   cout <<"Dinheiro: R$";
   cin >>pagamento;

   if (pagamento >= total){

      troco = pagamento - total;
      cout << "Troco: R$"<<troco<<endl;

   }

   else {

      do{
         troco = total - pagamento;
         cout << "O valor não confere com o total, restam mais: R$"<<troco<< endl;
         cout <<"Pague o restante: R$";
         cin >>pagamento;

         if (pagamento >= troco){
            troco = pagamento - troco;
            cout << "Troco: R$"<<troco<<endl;
            sai = true;

         }
         else {
            total = troco;
         }
      }while(sai == false);
   }
}
```

```
      else {
         cout << "Sua compra deu R$0,00. Obrigado!!!"<<endl;
      }

   }while (comeco == true);
   return 0;
}
```

18) Em uma eleição presidencial existem quatro candidatos. Os votos são informados por meio de código. Os códigos utilizados são: 1, 2, 3, 4 - Votos para os respectivos candidatos (você deve montar a tabela ex: 1 - Jose/ 2- João/etc) 5 - Voto Nulo 6 - Voto em Branco

Faça um programa que calcule e mostre: O total de votos para cada candidato; O total de votos nulos; O total de votos em branco; A percentagem de votos nulos sobre o total de votos; A percentagem de votos em branco sobre o total de votos.
Para finalizar o conjunto de votos tem-se o valor zero.

```
#include <iostream>

using namespace std;
int main(){
  int voto=1, bozo, lula, ciro, kelmon, nulo, branco, tot;

  do{
    cout<<"==================="<<endl;
    cout<<"   ELEICAO 2024"<<endl;
    cout<<"==================="<<endl;
    cout<<"CANDIDATOS:" <<endl;
    cout<<"[0] Sair"<<endl;
    cout<<"[1] Bolsonaro"<<endl;
    cout<<"[2] Lula    "<<endl;
    cout<<"[3] Ciro"<<endl;
    cout<<"[4] Padre Kelmon"<<endl;
    cout<<"[5] Voto nulo"<<endl;
    cout<<"[6] Voto em branco"<<endl;
    cout<<"==================="<<endl;
    cout<<"Insira seu voto: ";
    cin>>voto;
    if (voto != 0){
      tot++;
    }
    switch (voto){
      case 1:
        bozo++;
        break;
      case 2:
        lula++;
        break;
      case 3:
        ciro++;
        break;
      case 4:
        kelmon++;
        break;
      case 5:
        nulo++;
        break;
      case 6:
        branco++;
        break;
    }
```

```
    }while(voto != 0);
    cout<<"=========="<<endl;
    cout << "VOTOS"<<endl;
    cout<<"=========="<<endl;
    cout<<"Bolsonaro: "<<bozo<<" votos" <<endl;
    cout<<"Lula: "<<lula<<" votos" <<endl;
    cout<<"Ciro: "<<ciro<<" votos" <<endl;
    cout<<"Padre Kelmon: "<<kelmon<<" votos" <<endl;
    cout<<"Nulos: "<<nulo<<" votos" <<endl;
    cout<<"Em branco: "<<branco<<" votos" <<endl;
    nulo = (nulo*100)/tot;
    branco = (branco*100)/tot;
    cout<<"=============="<<endl;
    cout <<"PORCENTAGEM"<<endl;
    cout<<"=============="<<endl;
    cout<<"De "<<tot<<" votos"<<endl;
    cout<<nulo<<"% foram nulos" <<endl;
    cout<<"E "<<branco<<"% foram em branco" <<endl;

    return 0;
}
```

19) Faça uma rotina que remova um caracter de uma string do tipo char Str[100], dada a posição do caracter.

```
#include <iostream>

using namespace std;

void removerCaracter(string str, int posicao) {
   int tamanho = 0;
   string novaString;

   while (str[tamanho] != '\0') {
      tamanho++;
   }

   if (posicao <= 0 || posicao > tamanho) {
      cout << "Posição inválida!" << endl;
      return;
   }

   for (int i = 0; i < tamanho; i++) {
      if (i != posicao - 1) {
         novaString += str[i];
      }
   }

   cout << "String após remover o caractere na posição " << posicao << ": " <<
novaString << endl;
}

int main() {
   string str;
   int posicao;

   cout << "Informe a frase: ";
   cin >> str;
   cout << "Informe a posição de remoção: ";
   cin >> posicao;
   removerCaracter(str, posicao);

   return 0;
}
```

20) Faça uma rotina que insira um caracter em uma string do tipo char Str[100], dada a posição do caracter.

```
#include <iostream>
#include <cstring>

using namespace std;

void ins(char str[], int pos, char car) {
  int tam = strlen(str);

  if (pos < 0 || pos > tam) {
    cout << "Posicao Invalida" << endl;
    return;
}

for (int i = tam; i > pos; --i) {
  str[i] = str[i - 1];
}

str[pos] = car;

}

int main() {
  char str[100];
  char car;
  int pos;

  cout << "Digite uma frase: ";
  cin.getline(str,100);

  cout << "Digite o caractere a ser inserido: ";
  cin >> car;

  cout << "Digite a posicao onde o caractere sera colocado: ";
  cin >> pos;

  ins(str, pos, car);

  cout << "Frase Resultante: " << str << endl;

  return 0;
}
```

**Vetor**

21) Crie um programa que realize a busca por um valor específico em um vetor utilizando busca sequencial.

```
#include <iostream>

    using namespace std;

    int buscaSequencial(int vetor[], int tamanho, int valor) {
      for (int i = 0; i < tamanho; ++i) {
        if (vetor[i] == valor) {
          return i;
        }
      }
      return -1;
    }

    int main() {
      const int TAMANHO = 10;
      int vetor[TAMANHO];

      cout << "Digite 10 valores inteiros:" << endl;
      for (int i = 0; i < TAMANHO; ++i) {
        cout << "Digite o " << i+1 << "o valor: ";
        cin >> vetor[i];
      }

      int numero;
      cout << "Digite o numero que deseja encontrar: ";
      cin >> numero;

      int posicao = buscaSequencial(vetor, TAMANHO, numero);

      if (posicao != -1) {
        cout << "O numero " << numero << " foi encontrado na posicao " << posicao
    << " do vetor." << endl;
      } else {
        cout << "O numero " << numero << " nao foi encontrado no vetor." << endl;
      }

      return 0;
    }
```

22) Faça um programa que leia 10 números inteiros, armazene-os em um vetor, solicite um valor de referência inteiro e:
a) imprima os números do vetor que são maiores que o valor referência
b) retorne quantos números armazenados no vetor são menores que o valor de referência
c) retorne quantas vezes o valor de referência aparece no vetor.

```
#include <iostream>

using namespace std;

int main() {
   const int TAMANHO = 10;
   int vetor[TAMANHO];
   int valorReferencia;
   int maioresQueReferencia = 0;
   int menoresQueReferencia = 0;
   int ocorrenciasReferencia = 0;


   cout << "Digite 10 numeros inteiros:" << endl;
   for (int i = 0; i < TAMANHO; ++i) {
      cout << "Digite o " << i+1 << "o numero: ";
      cin >> vetor[i];
   }


   cout << "Digite o valor de referencia: ";
   cin >> valorReferencia;


   cout << "Numeros maiores que o valor de referencia:" << endl;
   for (int i = 0; i < TAMANHO; ++i) {
      if (vetor[i] > valorReferencia) {
         cout << vetor[i] << " ";
         maioresQueReferencia++;
      }
   }
   cout << endl;


   for (int i = 0; i < TAMANHO; ++i) {
      if (vetor[i] < valorReferencia) {
         menoresQueReferencia++;
      }
   }


   for (int i = 0; i < TAMANHO; ++i) {
      if (vetor[i] == valorReferencia) {
         ocorrenciasReferencia++;
      }
   }
```

```
    cout << "Quantidade de numeros menores que o valor de referencia: " <<
  menoresQueReferencia << endl;
    cout << "Quantidade de vezes que o valor de referencia aparece no vetor: " <<
  ocorrenciasReferencia << endl;

    return 0;
  }
```

23) Faça um programa que preencha um vetor de tamanho 100 com os 100 primeiros naturais que não são múltiplos de 7 ou que terminam com 7.

```
#include <iostream>

  using namespace std;

  int main() {
    const int TAMANHO = 100;
    int vetor[TAMANHO];
    int contador = 0;
    int numero = 1;


    while (contador < TAMANHO) {

      if (numero % 7 != 0 && numero % 10 != 7) {
        vetor[contador] = numero;
        contador++;
      }
      numero++;
    }


    cout << "Os primeiros 100 naturais que não são multiplos de 7 ou que terminam
  com 7 são:" << endl;
    for (int i = 0; i < TAMANHO; ++i) {
      cout << vetor[i] << " ";
    }
    cout << endl;

    return 0;
  }
```

24) Faça um programa para ler 10 números DIFERENTES a serem armazenados em um vetor. Os dados deverão ser armazenados no vetor na ordem que forem sendo lidos, sendo que caso o usuário digite um número que já foi digitado anteriormente, o programa deverá pedir para ele digitar outro número. Note que cada valor digitado pelo usuário deve ser pesquisado no vetor, verificando se ele existe entre os números que já foram fornecidos. Exibir na tela o vetor final que foi digitado.

```
#include <iostream>

using namespace std;


bool existeNoVetor(int vetor[], int tamanho, int numero) {
  for (int i = 0; i < tamanho; ++i) {
    if (vetor[i] == numero) {
      return true;
    }
  }
  return false;
}

int main() {
  const int TAMANHO = 10;
  int vetor[TAMANHO];
  int numero;
  int contador = 0;

  cout << "Digite 10 numeros diferentes:" << endl;


  for (int i = 0; i < TAMANHO; ++i) {
    cout << "Digite o " << i+1 << "o numero: ";
    cin >> numero;

    while (existeNoVetor(vetor, contador, numero)) {
      cout << "Numero repetido. Digite outro numero: ";
      cin >> numero;
    }

    vetor[contador] = numero;
    contador++;
  }

  cout << "Vetor final:" << endl;
  for (int i = 0; i < TAMANHO; ++i) {
    cout << vetor[i] << " ";
  }
  cout << endl;

  return 0;
```

```
}
```

**Matriz**

25) Crie um programa capaz de mostrar a transposta de uma matriz. A matriz deve ser lida pelo teclado.

```
#include <iostream>

using namespace std;


int main() {
   int linhas, colunas;


   cout << "Digite o numero de linhas da matriz: ";
   cin >> linhas;
   cout << "Digite o numero de colunas da matriz: ";
   cin >> colunas;


   int matriz[linhas][colunas];


   cout << "Digite os elementos da matriz:" << endl;
   for (int i = 0; i < linhas; ++i) {
      for (int j = 0; j < colunas; ++j) {
         cout << "Digite o elemento da posicao [" << i << "][" << j << "]: ";
         cin >> matriz[i][j];
      }
   }


   cout << "Matriz original:" << endl;
   for (int i = 0; i < linhas; ++i) {
      for (int j = 0; j < colunas; ++j) {
         cout << matriz[i][j] << " ";
      }
      cout << endl;
   }
```

```
    cout << "Transposta da matriz:" << endl;

    for (int j = 0; j < colunas; ++j) {

        for (int i = 0; i < linhas; ++i) {

            cout << matriz[i][j] << " ";

        }

        cout << endl;

    }

    return 0;

}
```

26) Leia uma matriz 10 x 10 e escreva a localização (linha e a coluna) do maior valor.

```
#include <iostream>

using namespace std;

int main() {
    const int LINHAS = 10;
    const int COLUNAS = 10;
    int matriz[LINHAS][COLUNAS];


    cout << "Digite os elementos da matriz 10x10:" << endl;
    for (int i = 0; i < LINHAS; ++i) {
        for (int j = 0; j < COLUNAS; ++j) {
            cout << "Digite o elemento da posicao [" << i << "][" << j << "]: ";
            cin >> matriz[i][j];
        }
    }


    int maiorValor = matriz[0][0];
    int linhaMaior, colunaMaior;

    for (int i = 0; i < LINHAS; ++i) {
        for (int j = 0; j < COLUNAS; ++j) {
            if (matriz[i][j] > maiorValor) {
                maiorValor = matriz[i][j];
                linhaMaior = i;
                colunaMaior = j;
            }
        }
    }


    cout << "O maior valor da matriz (" << maiorValor << ") está na linha " << linhaMaior
<< " e coluna " << colunaMaior << "." << endl;
```

```
    return 0;
}
```

27) Numa enfermaria existem quatro camas (cama 1, cama 2, cama 3 e cama 4) onde se encontram 4 pacientes a quem de hora a hora são medidas as pulsações ao longo de um dia (24 leituras do valor da pulsação para cada paciente). Desenvolva um algoritmo capaz de:

a. Proceder à leitura e armazenamento numa matriz de dimensão 24 x 4 dos valores das pulsações dos 4 pacientes ao longo das 24 horas de um dia.

b. Calcular e apresentar a média das pulsações para cada um dos pacientes.

c. Identificar a cama onde se encontra o paciente que apresentou maior valor médio das pulsações

```
#include <iostream>

using namespace std;

int main() {
    const int NUM_PACIENTES = 4;
    const int NUM_HORAS = 24;


    int pulsoes[NUM_HORAS][NUM_PACIENTES];


    cout << "Digite as pulsações dos pacientes ao longo de 24 horas:" << endl;
    for (int paciente = 0; paciente < NUM_PACIENTES; ++paciente) {
        cout << "Paciente " << paciente + 1 << ":" << endl;
        for (int hora = 0; hora < NUM_HORAS; ++hora) {
            cout << "Hora " << hora + 1 << ": ";
            cin >> pulsoes[hora][paciente];
        }
    }


    cout << "\nMedias das pulsacoes para cada paciente:" << endl;
    double mediaPaciente[NUM_PACIENTES] = {0};

    for (int paciente = 0; paciente < NUM_PACIENTES; ++paciente) {
        for (int hora = 0; hora < NUM_HORAS; ++hora) {
            mediaPaciente[paciente] += pulsoes[hora][paciente];
        }
        mediaPaciente[paciente] /= NUM_HORAS;
        cout << "Paciente " << paciente + 1 << ": " << mediaPaciente[paciente] << endl;
    }


    double maiorMedia = mediaPaciente[0];
    int camaMaiorMedia = 1;

    for (int paciente = 1; paciente < NUM_PACIENTES; ++paciente) {
```

```
      if (mediaPaciente[paciente] > maiorMedia) {
         maiorMedia = mediaPaciente[paciente];
         camaMaiorMedia = paciente + 1;
      }
   }

   cout << "\nO paciente com maior media de pulsacoes esta na cama " <<
camaMaiorMedia << "." << endl;

   return 0;
}
```

**Funções**

28) Crie uma função capaz de somar os elementos das linhas L1 e L2 de uma matriz. O resultado deve ser colocado na linha L2. Faça o mesmo com a multiplicação.

```
#include <iostream>


using namespace std;


const int NUM_COLUNAS = 5;


void somarLinhas(int matriz[][NUM_COLUNAS], int linha1, int linha2) {
   for (int j = 0; j < NUM_COLUNAS; ++j) {
      matriz[linha2][j] += matriz[linha1][j];
   }
}


void multiplicarLinhas(int matriz[][NUM_COLUNAS], int linha1, int linha2) {
   for (int j = 0; j < NUM_COLUNAS; ++j) {
      matriz[linha2][j] *= matriz[linha1][j];
   }
}


void exibirMatriz(int matriz[][NUM_COLUNAS]) {
   for (int i = 0; i < 3; ++i) {
      for (int j = 0; j < NUM_COLUNAS; ++j) {
         cout << matriz[i][j] << " ";
      }
      cout << endl;
   }
}


int main() {
```

```
    int matriz[3][NUM_COLUNAS] = {{1, 2, 3, 4, 5},
                                  {6, 7, 8, 9, 10},
                                  {11, 12, 13, 14, 15}};

    cout << "Matriz original:" << endl;
    exibirMatriz(matriz);

    somarLinhas(matriz, 0, 1);
    cout << "\nMatriz apos somar L1 com L2:" << endl;
    exibirMatriz(matriz);

    multiplicarLinhas(matriz, 0, 1);
    cout << "\nMatriz apos multiplicar L1 por L2:" << endl;
    exibirMatriz(matriz);

    return 0;
}
```

29) Faça uma função que retorne a posição de um dado caracter dentro de uma string.

```
#include <iostream>
#include <string>

using namespace std;

int encontrarCaractere(const string& str, char caractere) {
    for (int i = 0; i < str.length(); ++i) {
        if (str[i] == caractere) {
            return i;
        }
    }
    return -1;
}

int main() {
    string frase;
    char caractere;

    cout << "Digite uma frase: ";
    getline(cin, frase);

    cout << "Digite o caractere a ser encontrado: ";
    cin >> caractere;
```

```
    int posicao = encontrarCaractere(frase, caractere);

    if (posicao != -1) {
        cout << "O caractere '" << caractere << "' foi encontrado na posicao " << posicao
<< " da frase." << endl;
    } else {
        cout << "O caractere '" << caractere << "' nao foi encontrado na frase." << endl;
    }

    return 0;
}
```

30) Faça uma função que verifique se um valor é perfeito ou não. Um valor é dito perfeito quando ele é igual a soma dos seus divisores excetuando ele próprio. (Ex: 6 é perfeito, 6 = 1 + 2 + 3, que são seus divisores). A função deve retornar um valor booleano.

```
#include <iostream>

using namespace std;

bool ehPerfeito(int numero) {
    int somaDivisores = 0;

    for (int i = 1; i < numero; ++i) {
        if (numero % i == 0) {
            somaDivisores += i;
        }
    }

    return somaDivisores == numero;
}

int main() {
    int numero;

    cout << "Digite um numero para verificar se e perfeito: ";
    cin >> numero;

    if (ehPerfeito(numero)) {
        cout << "O numero " << numero << " e perfeito." << endl;
    } else {
        cout << "O numero " << numero << " nao e perfeito." << endl;
    }

    return 0;
}
```