

Técnicas de Programação

Prof. Protásio



Laboratório de Microengenharia/DEE/CEAR/UFPB

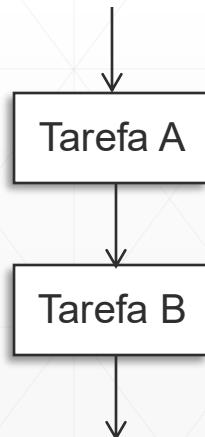
Programação Estruturada

- A base da programação estrutura é que um programa deve ser composto por **blocos de código** (**procedimentos**) que se interligam através de três **estruturas de controle**:
 - **Sequência**
 - **Seleção**, e
 - **Iteração**.

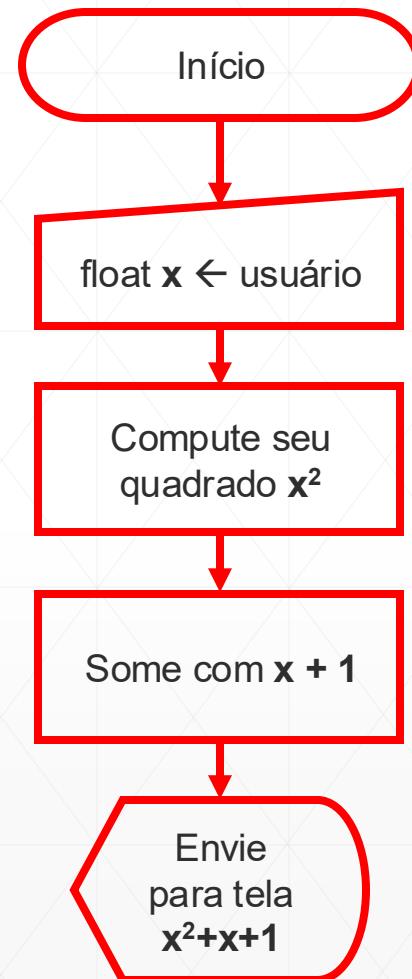


Sequência

- Passos de processamento ordenados de um programa.
- Exemplos:
 - “primeiro faça a *Tarefa A* e depois a *Tarefa B*”
- Pseudocódigo
 - :
 - **do Tarefa A**
 - **do Tarefa B**
 - :
- Em fluxograma



Codifique em C++ o fluxograma abaixo



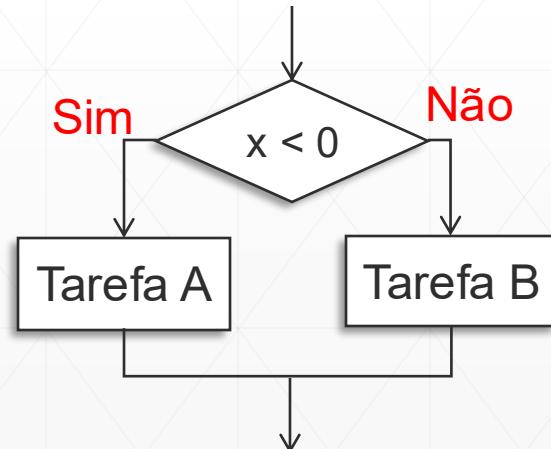
Estrutura de Controle de Seleção

- Seleciona o fluxo de execução do processamento baseado em ocorrências lógicas.
- A forma básica de seleção é pela construção IF

- Pseudocódigo

```
:  
if x < 0  
then Tarefa A  
else Tarefa B  
:  
:
```

- Em fluxograma



Estrutura de Controle de Seleção

▪ Sintaxe do comando IF

if (expressão) comando

if (expressão){
 Bloco de comandos
}

#include <iostream>
using namespace std;
int main() {
 float nota;
 cout << "Digite a nota do aluno" << endl;
 cin >> nota;
 if (nota >= 7)
 cout << "Aprovado" << endl;
 if (nota < 7) {
 cout << "Não aprovado" << endl;
 if (nota > 4)
 cout << "Final" << endl;
 if (nota <= 4)
 cout << "Reprovado" << endl;
 }
 return 0;
}

Exemplo

Situação-problema:

Altere o código para que, sendo o aluno aprovado, informe que ele é excelente se sua nota for acima de 9.5



Estrutura de Controle de Seleção

- Dupla condição IF-ELSE
- Sintaxe

if (expressão) comando1 else comando2



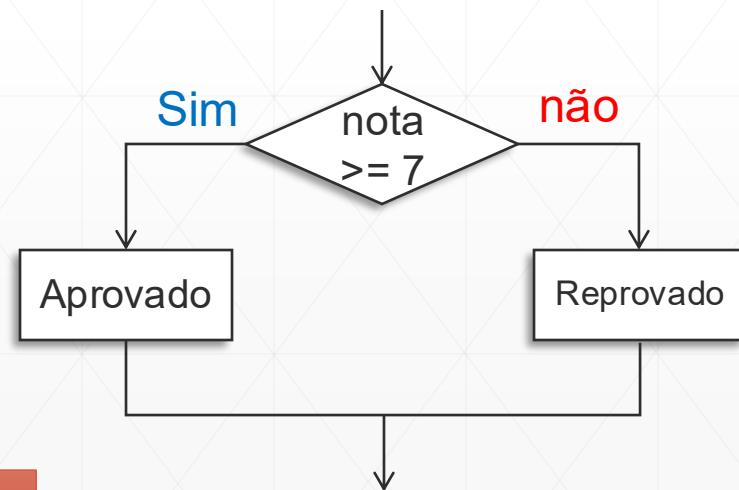
Se expressão verdadeira



Se expressão falsa

- Exemplo

```
if (nota >= 7)
    cout << "Aprovado" << endl;
else
    cout << "Reprovado" << endl;
```



Situação-problema:
Modifique o programa anterior
usando IF-ELSE

Estrutura de Controle de Seleção

- IF-ELSE aninhados
 - Testa várias condições em cadeia
 - Exemplo

```
if (x > 0)
    cout << "x eh positivo";
else if (x < 0)
    cout << "x eh negativo";
else
    cout << "x eh 0";
```

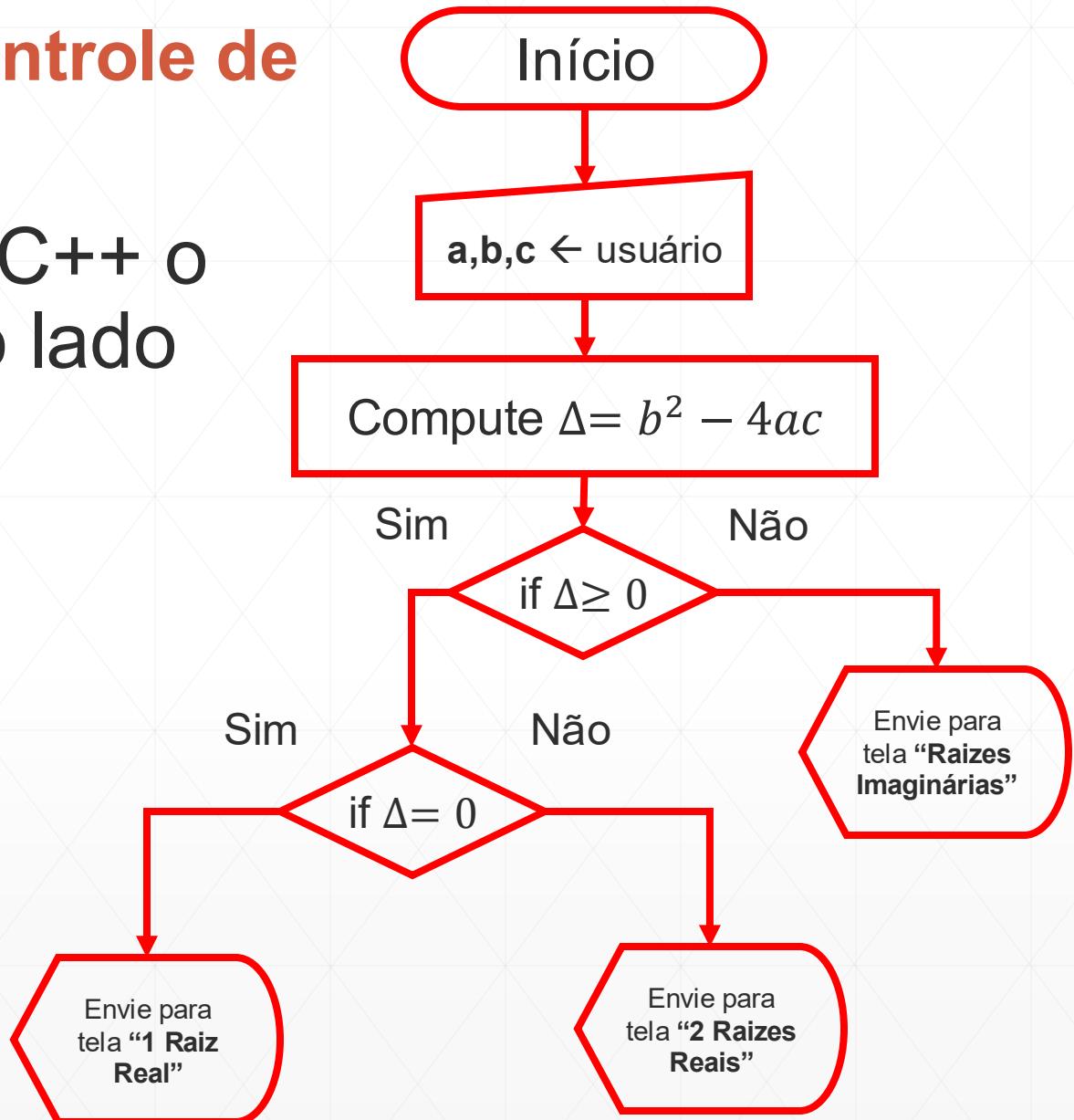
Situação-problema:

- Usando IF-ELSE aninhando, faça um programa que converta a nota numérica de um aluno para conceito:
 - 0→4: insuficiente
 - 4→7: abaixo da media
 - 7→9: acima da media
 - 9→10: excelente



Estrutura de Controle de Seleção

- Codifique em C++ o fluxograma ao lado



Estrutura de Controle de Seleção

- Estrutura switch-case
- Comparação com if

```
if (x == 1)
{
    cout << "x eh 1";
}
else if (x == 2) {
    cout << "x eh 2";
}
else {
    cout << "x tem outro valor";
}
```

```
switch (x) {
    case 1:
        cout << "x eh 1";
        break;
    case 2:
        cout << "x eh 2";
        break;
    default:
        cout << "x tem outro valor";
}
```

Estrutura de Controle de Seleção

■ Estrutura switch-case

■ Sintaxe

```
switch (expressão) {  
    case constante1:  
        grupo_de_comandos-1;  
        break;  
    case constante2:  
        grupo_de_comandos-2;  
        break; ...  
    default:  
        grupo_de_comandos_padrões  
}
```

```
#include <iostream>  
using namespace std;  
int main(){  
    int nota;  
    cout << "Digite a nota do aluno" << endl;  
    cin  >> nota;  
    switch (nota) {  
        case 1:  
            cout << "Reprovado" << endl;  
            cout << "Cuidado" << endl;  
            break;  
        case 2:  
            cout << "Reprovado" << endl;  
            break;  
        case 3:  
            cout << "Reprovado" << endl;  
            break;  
        case 4:  
            cout << "Reprovado" << endl;  
            break;  
        case 5:  
            cout << "Final" << endl;  
            break;  
        case 6:  
            cout << "Final" << endl;  
            break;  
        case 7:  
            cout << "Aprovado" << endl;  
            cout << "Por pouco" << endl;  
            break;  
        case 8:  
            cout << "Aprovado" << endl;  
            break;  
        case 9:  
            cout << "Aprovado" << endl;  
            break;  
        case 10:  
            cout << "Aprovado" << endl;  
            cout << "Showww" << endl;  
            break;  
        default:  
            cout << "Erro: digite valor entre 0 e 10";  
    }  
}
```

Estrutura de Controle de Seleção

■ Estrutura switch-case

```
#include <iostream>
using namespace std;
int main(){
    int opcao;
    cout << "Digite um numero" << endl;
    cin >> opcao;
    switch (opcao) {
        case 0:
        case 1:
        case 2:
            cout << "Menor que 3";
            break;
        case 3:
            cout << "Igual a 3";
            break;
        default:
            cout << "Maior que 3 ";
    }
    return 0;
}
```

Exemplo 1

```
#include <iostream>
using namespace std;
int main(){
    char temp;
    cout << "Digite uma letra" << endl;
    cin >> temp;
    switch (temp) {
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
            cout << "Eh uma vogal minúscula";
            break;
        default:
            cout << "Nao eh uma vogal minúscula";
    }
    return 0;
}
```

Exemplo 2

Estrutura switch-case

■ Construção de menu

```
#include <iostream>
using namespace std;
void sub(void);
void mul(void);

int main() {
    cout << "UNIVERSIDADE FEDERAL DA PARAIBA" << endl;
    cout << "CEAR - Departamento de Engenharia Eletrica" << endl;
    cout << "Tecnicas de Programacao: Menu principal" << endl << endl;
    cout << "Opcoes: (1) Soma, (2) Subtracao, (3) Multiplicacao e (0) Sair do programa." << endl;
    cout << "Digite a opcao desejada: ";
    int opcao;
    cin >> opcao;
    switch (opcao) {
        case 1:
            int x,y;
            cout << "Soma de dois valores. Digite os valores:" << endl;
            cin >> x >> y;
            cout << "O resultado eh " << x + y << endl;
            break;
        }
        case 2:
            sub();
            break;
        case 3:
            mul();
            break;
        case 0:
            cout << "Fim do programa!" << endl;
            return 0;
        default:
            cout << "Opcao invalida" << endl;
        }
    cout << "Fim do programa!" << endl;
    return 0;
}

void sub(void) {
    int x,y;
    cout << "Subtracao de dois valores. Digite os valores:" << endl;
    cout << "O resultado eh " << x - y << endl;
}

void mul(){
    int x,y;
    cout << "Multiplicacao de dois valores. Digite os valores:" << endl;
    cout << "O resultado eh " << x * y << endl;
}
```

Estrutura de Controle de Iteração

- Execução **repetitiva** de bloco de códigos do programa.
- A forma básica de iteração é pela construção **WHILE/DO** ou **DO/WHILE**

WHILE/DO

- Pseudocódigo

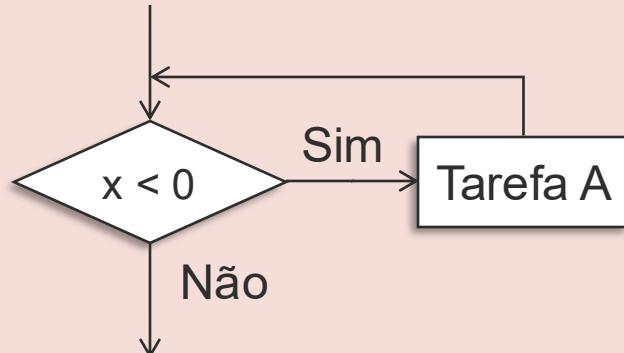
:

while $x < 0$

do Tarefa A

:

- Fluxograma



DO/WHILE

- Pseudocódigo

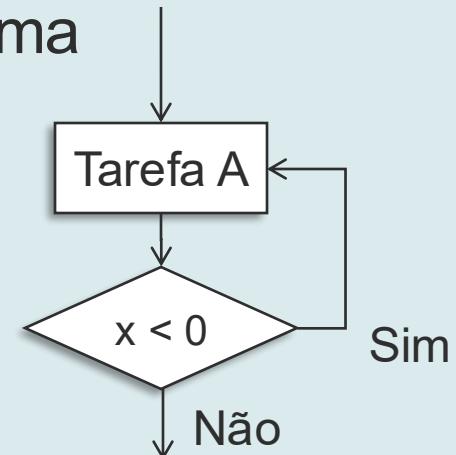
:

do Tarefa A

while $x < 0$

:

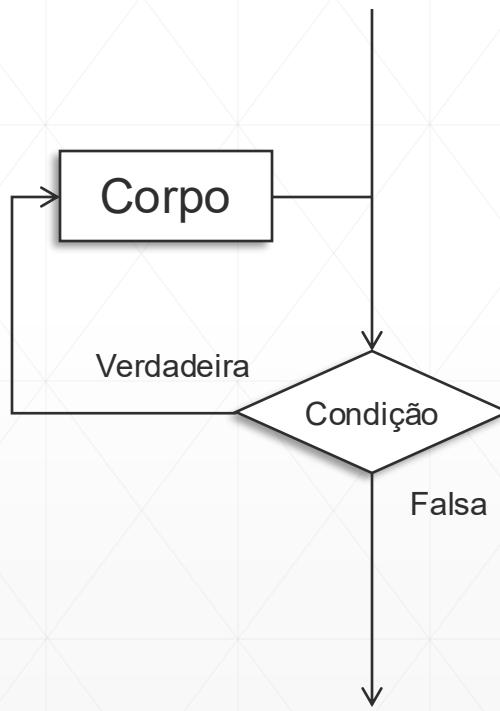
- Fluxograma



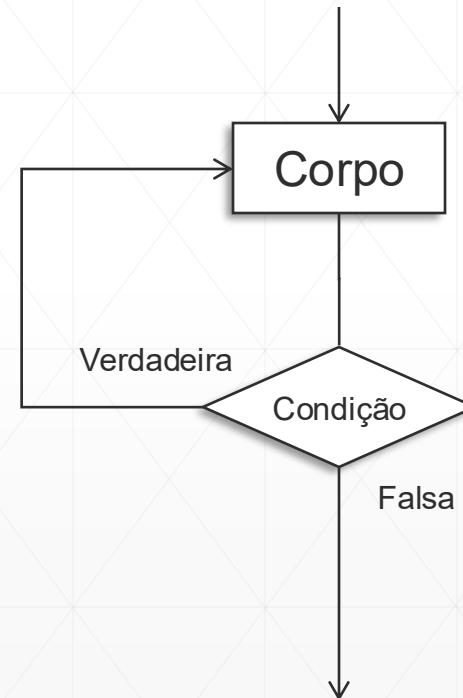
Estrutura de Controle de Iteração

- Laço (*loop*)

TESTA → EXECUTA



EXECUTA → TESTA



Estrutura de Controle de Iteração

- Laço (*loop*)

- Sintaxe

TESTA → EXECUTA

while (*expressão*) comando

- Exemplos

```
#include <iostream>

using namespace std;
int main () {
    int x = 10;
    while (x>0) {
        cout << x << endl;
        --x;
    }
    return 0;
}
```

EXECUTA → TESTA

do comando **while** (*expressão*)

```
#include <iostream>
#include <string>
using namespace std;
int main () {
    string texto;
    do {
        cout << "Digite um texto: ";
        getline (cin, texto);
        cout << "Voce digitou: " << texto << '\n';
    } while (texto != "bye");
```

Estrutura de Controle de Iteração

■ Laço while (*true*)

```
#include <iostream>

using namespace std;

int main () {
    int contador = 0;
    while (true) {
        cout << contador << endl;
        contador++;
    }
}
```

Situação-problema:

- Modifique o programa para que este receba um valor 0 ou 1 do usuário e se for 1 o laço seja executado.

Estrutura de Controle de Iteração

▪ Laço while (*true*)

Laço infinito

```
#include <iostream>

using namespace std;

int main () {
    int contador = 0;
    while (true) {
        cout << contador << endl;
        contador++;
    }
    cout << "Fim" << endl;
}
```

Laço finito

```
#include <iostream>

using namespace std;

int main () {
    int contador = 0;
    while (true) {
        cout << contador << endl;
        contador++;
        if (contador > 10)
            break;
    }
    cout << "Fim" << endl;
}
```



Estrutura de Controle

■ Laço while (*parte 2*)

Uso do **break**

```
#include <iostream>

using namespace std;

int main () {
    int contador = 0;
    while (true) {
        contador++;
        if (contador == 10)
            break;
        cout << contador << endl
    }
    cout << "Fim" << endl;
}
```

Situação-problema 2:

- Modifique o programa para que pular a impressão de número pares.

Uso do **continue**

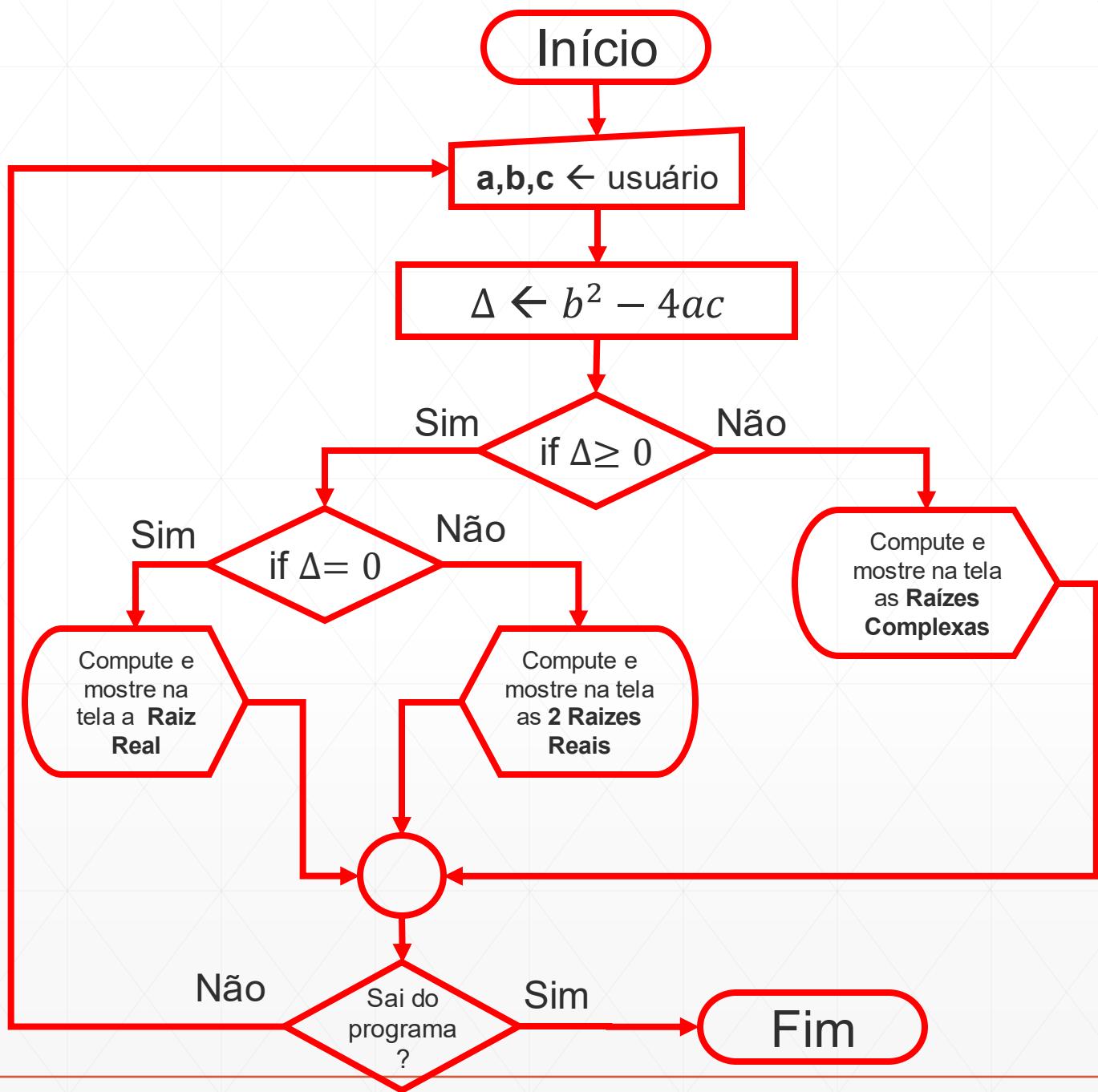
```
#include <iostream>

using namespace std;

int main () {
    int contador = 0;
    while (true) {
        contador++;
        if (contador == 10)
            continue;
        cout << contador << endl;
    }
    cout << "Fim" << endl;
}
```

Situação-problema 1:

- Modifique o programa para que pular a impressão entre 10 e 20.



Estrutura switch-case

- Construção de menu com retorno com início da opções

```
#include <iostream>
using namespace std;
void sub(void);

int main(){
    cout << "UNIVERSIDADE FEDERAL DA PARAIBA" << endl;
    cout << "CEAR - Departamento de Engenharia Eletrica" << endl;
    cout << "Tecnicas de Programacao: Menu principal" << endl << endl;

    while (true) {
        cout << "Opcoes: (1) Soma, (2) Subtracao e (0) Sair do programa." << endl;
        cout << "Digite a opcao desejada: ";
        int opcao;
        cin >> opcao;
        switch (opcao) {
            case 1:{
                int x,y;
                cout << "Soma de dois valores. Digite os valores:" << endl;
                cin >> x >> y;
                cout << "O resultado eh " << x + y << endl;

            }break;
            case 2:
                sub();
                break;
            case 0:
                cout << "Fim do programa!" << endl;
                return 0;
            default:
                cout << "Opcao invalida" << endl;
        }
    }
    return 0;
}

void sub(void){
    int x,y;
    cout << "Subtracao de dois valores. Digite os valores:" << endl;
    cin >> x >> y;
    cout << "O resultado eh " << x - y << endl;
}
```

Estrutura de Controle de Iteração

■ Laço FOR

```
for (inicialização; condição; incremento)
```

```
#include <iostream>
using namespace std;
int main () {
    for (int x=0; x<10; x++)
        cout << x << endl;

    cout << "Fim" << endl;
    return 0;
}
```

Equivalentе

```
#include <iostream>

using namespace std;
int main () {
    int x = 0;
    while (x<10) {
        cout << x << endl;
        x++;
    }

    cout << "Fim" << endl;
    return 0;
}
```

Estrutura de Controle de Iteração

■ Laço FOR

```
for (inicialização; condição; incremento)
```

```
#include <iostream>
using namespace std;
int main () {
    float somatorio = 0; // Calcule o somatório de 10 valores dados pelo usuário
    float temp;
    for (int i=0; i<10; i++){
        cout << "Digite " << i + 1 << "º valor: ";
        cin >> temp;
        somatorio = somatorio + temp;
        cout << "Somatório atual = " << somatorio << endl;
    }
    cout << "Somatório Final = " << somatorio << endl;
    return 0;
}
```

} Grupo de comandos



Estrutura de Controle de Iteração

■ Laço FOR

- Geração de uma sequência pseudoaleatória de **valores inteiros**

```
#include <iostream>
#include <cstdlib> /* srand, rand */
#include <ctime> /* time */
using namespace std;

int main () {
    int dado;
    srand( time(NULL) ); // inicializa a semente(seed)
    for (int x=0; x<10; x++){
        dado = rand()%6 + 1; // Gera um valor pseudo-aleatório
        // entre 1 e 6
        cout << dado << endl;
    }
    return 0;
}
```

time(NULL) retorna a quantidade de segundos desde 00:00 hora de 1 de Jan de 1970 UTC

Faça um programa que emule um sorteio da MEGA-SENA, que gere 6 números entre 1 e 60



Estrutura de Controle de Iteração

- Laço FOR
 - Geração de uma sequência pseudoaleatória de valores ponto flutuante

```
#include <iostream>
#include <cstdlib> /* srand, rand */
#include <ctime> /* time */
using namespace std;

int main(){
    double aleatorio, media, max = 0, min = 1;
    srand( time(NULL) ); // inicializa a semente(seed)
    for (int x=0; x<1000; x++){
        aleatorio = (double)rand()/RAND_MAX; // Gera um valor pseudo-
                                                // aleatório entre 0 e 1
        cout << aleatorio << endl;
        media = media + aleatorio;
        if (aleatorio > max) max = aleatorio;
        if (aleatorio < min) min = aleatorio;
    }
    cout << "Média = " << media/1000 << endl;
    cout << "Max = " << max << endl;
    cout << "Min = " << min << endl;
    return 0;
}
```



Estrutura de Controle de Iteração

■ Laço infinito com for

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

int main () {
    for (;;) {
        int x;
        cout << "Digite um numero:" << endl;
        cin >> x;
    }
    cout << "Fim" << endl;
}
```

Resultado

10, 9, 8, 7, 6, 4, 3, 2, 1, Fim

```
#include <iostream>
using namespace std;
int main () {
    for (;;) {
        int x;
        cout << "Digite um numero:" << endl;
        cin >> x;
        if (x == 0)
            break;
    }
    cout << "Fim" << endl;
}
```

Uso do break

```
#include <iostream>          Uso do continue
using namespace std;
int main () {
    for (int x=10; x>0; x--)
    {
        if (x==5)
            continue;
        cout << x << ", ";
    }
    cout << "Fim" << endl;
}
```

Pula o restante
do laço na
iteração corrente
e volta para o
início do laço.



Estrutura de Controle de Iteração

- Múltiplas inicializações

```
#include <iostream>

using namespace std;

int main () {

    for (int n=0, i=100 ; n!=i ; ++n, --i ){
        cout << n << ", " << i << endl;
    }
    cout << "Fim" << endl;
}
```



Lista de Exercícios

1. **Construindo um menu**, elabore um código que o usuário tenha opção de **escolher**, na tela inicial, quais das operações abaixo será executada e ter uma **opção de retornar à tela inicial ou sair do programa** (use a estrutura *switch-case*).
 - I. Solicite ao usuário duas letras do alfabeto (que podem ser tanto maiúsculas quanto minúsculas) e mostre na tela a sequência completa de letras entre essas duas letras.
 - II. Mostre na tela o valor de **y** do somatório $y = \sum_{x=a}^b (x + 3)$, em que os valores de **a** e **b** inteiros sejam dados pelo usuário.
 - III. Mostre na tela a **raiz quadrada** de todos os números inteiros situados no intervalo **a** e **b**, sendo **a** e **b** dados pelo usuário.
 - IV. Mostre na tela o **padrão numérico/asteriscos** abaixo de acordo com o valor de **n**, dado pelo usuário (no exemplo abaixo, **n = 6**):
 - 123456
 - 12345*
 - 1234**
 - 123***
 - 12****
 - 1*****
 - V. Mostre na tela os valores de $f(x)$ no intervalo $[x_{\min}, \dots, x_{\max}]$ com passo **δ**] da função $f(x) = ax^2 + bx + c$, em que os valores de **x_{min}**, **x_{max}**, **δ**, **a**, **b** e **c** sejam dados pelo usuário. Todos os valores envolvidos devem ser em ponto flutuante.
 - VI. Solicite ao usuários os valores de **a**, **b** e **c** de uma equação de 2º grau na forma **ax²+bx+c** e mostre nas tela **os zeros reais e imaginários**, caso tenham.