

Aula 8 – Comandos de Seleção

Algoritmos e Programação de Computadores

Profs: Ronaldo Castro de Oliveira – ronaldo.co@ufu.br

Anilton Joaquim da Silva – anilton@ufu.br

Introdução

- Até agora não aprendemos a dizer para o computador “Se for assim, então faça assado! Senão, faça cozido!”
- Como exemplo no calculo das raizes reais do segundo grau temos que verificar o valor calculado do DELTA:
 - Se $\text{DELTA} < \text{ZERO}$, então não existe raizes reais par serem calculadas
 - Se $\text{DELTA} == \text{ZERO}$, então as raizes são iguais
 - Se $\text{DELTA} > \text{ZERO}$, então existem duas raizes reais e as mesmas devem ser calculadas
- As instruções de seleção utilizam dos operadores **relacionais** e **lógicos** para definição de qual caminho deve ser seguido na execução.

Operadores

- Relacionais

== igual a
!= diferente de
> maior que
< menor que
>= maior ou igual a
<= menor ou igual a

- Lógicos

&& and
|| ou
! not

A	!A
V	F
F	V

A	B	A && B
V	V	V
V	F	F
F	V	F
F	F	F

A	B	A B
V	V	V
V	F	V
F	V	V
F	F	F

Comando if-else

- Em C, testes simples (tudo o que você realmente precisa) podem ser feitos com a estrutura if, que tem uma das seguintes sintaxes:
 - **if (expressão lógica)**
bloco de comandos 1
 - **if (expressão lógica)**
bloco de comandos 1
else
bloco de comandos 2

Comando if-else

- Uma expressão lógica é uma expressão cuja avaliação resulte em **verdadeiro** ou **falso** como, por exemplo, as expressões que usam os operadores relacionais
- Um bloco de comandos é ou uma instrução ou um conjunto de instruções dentro de { }.
- Quando a expressão lógica é avaliada, se seu resultado for verdadeiro, então o bloco de comandos 1 será executado. Se o resultado for falso, o bloco de comandos 2 será executado.

OBS: Se no comando de decisão não existir o **else** o programa continua execução na próxima linha de comando.

Exemplo: Raizes de Eq de 2º grau

```
#include <iostream>
#include <cstdlib> // inclui o system("pause")
#include <math.h> // funções matemáticas
using namespace std;

//Função que calcula o DELTA de uma eq de 2 grau
float calcula_delta(float A, float B, float C)
{
    return (pow(B, 2) - 4*A*C);
}

//Função que imprime as raizes reais do 2 grau
void imprime_raizes (float A, float B, float C)
{
    float D;
    D = calcula_delta( A, B, C);
    if (D < 0)
    {
        cout << "Não existem raizes reais!" << endl;
    }
    else if (D == 0)
    {
        cout << "A equacao tem duas raizes iguais!" << endl;
        cout << "X1 e X2 = " << (-B/(2*A));
    }
    else
    {
        cout << "A equação tem duas raizes distintas!" << endl;
        cout << "X1 = " << (-B + sqrt(D))/(2*A) << endl;
        cout << "X2 = " << (-B - sqrt(D))/(2*A) << endl;
    }
}

//Função principal
int main()
{
    int A, B, C;
    float DELTA, X1, X2;
    cout << "Calculo raizes da eq do 2o grau - AX2 + BX + C = 0."
        << endl;
    // exemplo: X2 - 5*X +6 = 0   X1 = 2   X2 = 3

    cout << "Digite o valor de A: ";
    cin >> A;
    cout << "Digite o valor de B: ";
    cin >> B;
    cout << "Digite o valor de C: ";
    cin >> C;

    cout << "DELTA = " << calcula_delta(A, B, C);
    imprime_raizes(A, B, C);

    return 0;
}
```

Funções que não retornam valores

Procedimentos

- A função `imprime_raizes` tem por objetivo imprimir na tela as raízes da equação de segundo grau, se existirem. Esta função não tem o objetivo de calcular nada mas somente apresentar resultados na tela não necessitando retornar nenhum valor.
- Funções sem resultado são denominadas procedimentos e, em C, são declaradas como qualquer outra função, apenas com uma particularidade: o tipo do resultado é do tipo **void** e não é necessário incluir o comando **return** no final.

Funções que não retornam valores

Procedimentos

- **Como ficaria a função:**

```
//Função que imprime as raizes reais do 2 grau
void imprime_raizes (float A, float B, float C)
{
    float D;
    int qtd;

    D = calcula_delta( A, B, C);
    qtd = raizes ( A, B, C);
    if (qtd ==0)
    {
        cout << "Não existem raizes reais!" << endl;
    }
    else if (qtd == 1)
    {
        cout << "A equacao tem duas raizes iguais!" << endl;
        cout << "X1 e X2 = " << (-B/(2*A));
    }
    else
    {
        cout << "A equação tem duas raizes distintas!" << endl;
        cout << "X1 = " << (-B + sqrt(D))/(2*A) << endl;
        cout << "X2 = " << (-B - sqrt(D))/(2*A) << endl;
    }
}
```


Tipo primitivo **bool**

- O `if` avalia uma expressão lógica que resultam em verdadeiro ou falso. Na linguagem C(++), quaisquer números inteiros podem também ser avaliados como verdadeiro ou falso, seguindo a seguinte regra:
 - **0** (zero) corresponde a falso.
 - **1** (um) ou qualquer outro número corresponde a verdadeiro
- Em C++, também é possível utilizar os valores **true** e **false**, que correspondem, respectivamente, a **1** e **0**. Estes dois valores compõem o conjunto dos booleanos, ou melhor, o tipo primitivo `bool`

Tipo primitivo **bool**

- Escreva um programa que leia um ano, chame uma função para verificar se o ano é bissexto e imprima sim ou não.

```
#include <iostream>
```

```
using namespace std;
```

```
bool bissexto ( int ano)
```

```
{  
    if(((ano % 4 == 0) && (ano % 100 != 0)) ||  
        (ano % 400 == 0))  
    {  
        return true;  
    }  
    else  
    {  
        return false;  
    }  
}
```

```
int main()
```

```
{  
    int ano;  
    cout << "Digite o ano que deseja verificar se e  
        bissexto: ";  
    cin >> ano;  
    cout << "O ano " << ano;  
    if(bissexto(ano))  
    {  
        cout << " e bissexto" << endl;  
    }  
    else  
    {  
        cout << " nao e bissexto" << endl;  
    }  
    return 0;  
}
```

Prioridade de operadores

- Para operadores lógicos:
 - primeiro o NOT (!), depois o AND (&&) e por último o OR (||)
- Expressões com vários tipos de operadores:
 1. Parênteses, incremento e decremento (++ , --)
 2. not (!)
 3. Multiplicação, divisão e módulo (o que aparecer primeiro);
 4. Soma e subtração;
 5. Operadores relacionais (<, <=, >, >=)
 6. Operadores relacionais (==, !=)
 7. and (&&)
 8. or (||)
 9. Atribuição (=, +=, -=, *=, /=, %=)

Comando **switch**

- Seja um programa que leia o valor numérico de um mês e imprima o nome do mês correspondente:

```
...
int mes;
cout << "Digite o numero do mês: " << endl;
cin >> mes;
...
if (mês == 1)
    cout << "Janeiro" << endl;
if (mês == 2)
    cout << "Fevereiro" << endl;
...
if (mês == 12)
    cout << "Dezembro" << endl;
...
```

Comando **switch**

- Em vez de usar vários if e else-if, uma solução melhor seria usar switch, criado exatamente para tratar estas situações. A sintaxe do uso do switch é a seguinte:

```
switch (identificador)
{
    case valor1: bloco_comandos1
    case valor2: bloco_comandos2
    . . .
    case valorN: bloco_comandosN
    default: bloco_comandos_default
}
```

Comando switch

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

void imprime_mes(int mes)
{
    switch (mes)
    {
        case 1:
            cout << "Janeiro";
            break;
        case 2:
            cout << "Fevereiro";
            break;
        case 3:
            cout << "Marco";
            break;
        case 4:
            cout << "Abril";
            break;
        case 5:
            cout << "Maio";
            break;
        case 6:
            cout << "Junho";
            break;
        case 7:
            cout << "Julho";
            break;
        case 8:
            cout << "Agosto";
            break;
```

```
        case 9:
            cout << "Setembro";
            break;
        case 10:
            cout << "Outubro";
            break;
        case 11:
            cout << "Novembro";
            break;
        case 12:
            cout << "Dezembro";
            break;
    }
}

int main()
{
    int dia, mes, ano;

    cout << "Entre com a data: " << endl;
    cout << "Dia? " << endl;
    cin >> dia;
    cout << "Mes? " << endl;
    cin >> mes;
    cout << "Ano? " << endl;
    cin >> ano;

    cout << " A data e: " << dia << " de ";
    imprime_mes(mes);
    cout << " de " << ano << endl;

    return 0;
}
```

Exercício - Menu

- Implemente uma função chamada menu que imprima o seguinte menu na tela:
 - 1. Soma
 - 2. Média
 - 3. Menor
 - 4. Maior
- Leia o valor selecionado e retorne o número da opção escolhida.
- Implemente a função main de forma a ler três números quaisquer e então chame a função menu definida acima para decidir o que fazer. O resultado da função deve ser armazenando em uma variável e seu conteúdo testado com switch. Cada opção deve invocar a função respectiva, que calculará e retornará o que se pede. A função main imprimirá então o resultado.

Exercício - Menu

```
#include <iostream>

using namespace std;

int menu()
{
    int opcao;
    cout << "Escolha uma das opções abaixo: " << endl;
    cout << "1 - Soma" << endl << "2 - Media" << endl
         << "3 - Menor" << endl << "4 - Maior" << endl;
    cin >> opcao;
    return opcao;
}

int menor(int x, int y, int z)
{
    if(x <= y && x <= z)
        return x;
    if(y <= x && y <= z)
        return y;
    return z;
}
```

```
int maior(int x, int y, int z)
{
    if(x >= y && x >= z)
        return x;
    else if(y >= x && y >= z)
        return y;
    else
        return z;
}

int soma(int x, int y, int z)
{
    return x+y+z;
}

float media(int x, int y, int z)
{
    float somatorio;
    somatorio = soma(x,y,z);
    return (somatorio / 3.0);
}
```


Exercício - Menu

```
int main()
{
    int a, b, c;
    int opcao;
    cout << "Digite tres numero inteiros: " << endl;
    cout << "Numero 1: ";
    cin >> a;
    cout << "Numero 2: ";
    cin >> b;
    cout << "Numero 3: ";
    cin >> c;

    opcao = menu();
    switch(opcao)
    {
        case 1:
            cout << "A soma dos tres numeros eh " << soma(a,b,c);
            break;
        case 2:
            cout << "A media dos tres numeros eh " << media(a,b,c);
            break;
        case 3:
            cout << "O menor dentre os tres numeros eh " << menor(a,b,c);
            break;
        case 4:
            cout << "O maior dentre os tres numeros eh " << maior(a,b,c);
            break;
        default:
            cout << "Opcao invalida. Execute o programa novamente e leia direito as opcoes.";
    }
    return 0;
}
```