Estruturas Condicionais

Aula – Tópico 3

Problema 5

- Determine as raízes da equação $ax^2 + bx + c = 0$.

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char *argv[])
{
  int a = 2, b = 3, c = 1;
  float delta,x1,x2;

  delta = b*b - 4*a*c;
  printf("A equacao %s\n", (delta >=0) ? "possui raizes reais" :
                                         "nao possui raizes reais");
  if (delta >= 0)
  {
    printf("As raizes sao %s\n", (delta > 0)? "diferentes" : "iguais");
    x1 = (-b + sqrt(delta))/(2*a);
    x2 = (-b - sqrt(delta))/(2*a);
    printf("Raiz x1 = %f\n", x1);
    printf("Raiz x2 = %f\n", x2);
  }

  system("PAUSE");
  return 0;
}
```

Processamento condicional

- Todo programa na linguagem C inicia sua execução na primeira instrução da função main.
- As instruções são executadas sequencialmente, na ordem em que aparecem no texto.
- Muitas vezes, é necessário executar um conjunto de instruções se uma condição for verdadeira e, caso contrário, um outro conjunto de instruções.
- Quando um programa executa ou deixa de executar instruções com base no valor de uma condição, o programa realiza um processamento condicional.

Processamento condicional

- O programa p05.c realiza um processamento condicional.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char *argv[])
{
  int a = 2, b = 3, c = 1;
  float delta,x1,x2;

  delta = b*b - 4*a*c;
  printf("A equacao %s\n", (delta >=0) ? "possui raizes reais" :
                                         "nao possui raizes reais");

  if (delta >= 0)
  {
    printf("As raizes sao %s\n", (delta > 0)? "diferentes" : "iguais");
    x1 = (-b + sqrt(delta))/(2*a);
    x2 = (-b - sqrt(delta))/(2*a);
    printf("Raiz x1 = %f\n", x1);
    printf("Raiz x2 = %f\n", x2);
  }

  system("PAUSE");
  return 0;
}
```

Estas instruções serão executadas **somente se** delta >= 0.

Processamento condicional

- Para executar um processamento condicional, um programa precisa utilizar o comando if.
- Todo comando if requer uma condição. O valor de uma condição pode ser verdadeiro ou falso.
- Em C, não existe um tipo de dados específico para representar valores lógicos (V ou F).
- Qualquer valor diferente de zero é interpretado como verdadeiro, enquanto zero é falso.

Operadores relacionais

- Para escrever condições, são utilizados os operadores relacionais e os operadores lógicos.

Operador	Significado
>	Maior do que.
<	Menor do que.
>=	Maior do que ou igual a.
<=	Menor do que ou igual a.
==	Igual a.
!=	Diferente de.

Condição	Valor lógico
(a != x)	Verdadeiro.
(a/2.0 == x)	Verdadeiro.
(a/2 == x)	Falso.
(a/x < 2)	Falso.
(a)	Verdadeiro.
(a – 2*x)	Falso.

```
int a = 3; float x = 1.5;
```

Operadores lógicos

- Os operadores lógicos permitem combinar várias condições em uma única expressão lógica.

Operador	Significado
&&	Conjunção lógica (“and”)
||	Disjunção lógica (“or”)
!	Negação lógica (“not”)

Expressão	Valor Lógico
((a/2 == x) && (a > 2))	Falso.
((x <= a) && (a >= 2*x))	Verdadeiro.
(!(a/3 <= x))	Falso.
(a && x)	Verdadeiro.
((a – 2*x) || (x < a/2))	Falso.

```
int a = 3; float x = 1.5;
```

Operador condicional

- O operador condicional na linguagem C tem a seguinte sintaxe:

```
(condição) ? resultado-se-condição-verdadeira : resultado-se-condição-falsa
```

- Os resultados podem ser de qualquer tipo (int, float, char, double) e mesmo strings.

- Exemplos:

```
(b != 0) ? a/b : 0
(peso <= 75) ? "ok" : "deve emagrecer"
```

Operador condicional

- O operador condicional pode ser usado em atribuições.

- Exemplo:

```
float nota1 = 5.0, nota2 = 4.0;

media = ((nota1 >= 3) && (nota2 >= 5)) ?
        (nota1 + 2*nota2)/3 :
        (nota1 + nota2)/2;
```

media recebe o valor 4.5

Qual seria o valor de média se:

```
float nota1 = 5.0;
float nota2 = 6.5;
```

Operador condicional

- No programa p05.c, o operador condicional é usado dentro da função printf.

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char *argv[])
{
  int a = 2, b = 3, c = 1;
  float delta,x1,x2;

  delta = b*b - 4*a*c;
  printf("A equacao %s\n", (delta >=0) ? "possui raizes reais" :
                                         "nao possui raizes reais");
  if (delta >= 0)
  {
    printf("As raizes sao %s\n", (delta > 0)? "diferentes" : "iguais");
    x1 = (-b + sqrt(delta))/(2*a);
    x2 = (-b - sqrt(delta))/(2*a);
    printf("Raiz x1 = %f\n", x1);
    printf("Raiz x2 = %f\n", x2);
  }

  system("PAUSE");
  return 0;
}
```

Atribuição e teste de igualdade

- Atenção!
  - Um erro comum em linguagem C é usar o operador de atribuição (=) em vez do operador relacional (==) em condições que testam igualdade.

```
int fator = 3;
if (fator == 1)
{
   printf("O fator e' unitario\n");
}
printf("fator = %d\n", fator)
```

Imprime:
fator = 3
pois:
(fator == 1) é falso!

```
int fator = 3;
if (fator = 1)
{
   printf("O fator e' unitario\n");
}
printf("fator = %d\n", fator)
```

Imprime:
O fator e' unitario
fator = 1
pois:
(fator = 1) é verdadeiro!

Comando if-else

- Todo comando if requer uma condição que pode ser verdadeira ou falsa.

- Caso a condição seja verdadeira, o comando if executa um conjunto de instruções, podendo deixar de executar um outro conjunto alternativo.

- Quando existe um conjunto de instruções a ser executado, caso o valor da condição seja falso, utiliza-se o comando if-else.

Comando if-else

- Exemplo:

```
if (delta >=0)
{
      x1 = (-b + sqrt(delta))/(2*a);
      x2 = (-b - sqrt(delta))/(2*a);
}
else
{
      printf("Sem raízes reais.");
}
```

- Um conjunto de instruções começa com o símbolo { e termina com o símbolo }. Caso, o conjunto contenha apenas uma instrução, as chaves são opcionais.

Comando if-else

- Qualquer instrução pode fazer parte de um conjunto de instruções, inclusive um comando if ou um comando if-else.

```
if (delta >=0)
{
      x1 = (-b + sqrt(delta))/(2*a);
      if (delta == 0)
            x2 = x1;
      else
            x2 = (-b - sqrt(delta))/(2*a);
}
else
{
      printf("Sem raízes reais.");
}
```

Por que não foram usadas as chaves { } neste comando?

A importância dos recuos

- Programas mais complexos são mais difíceis de ler e compreender.
- Uma forma de melhorar a legibilidade do programa é usar recuos.
- Os recuos devem ser usados sempre após o símbolo {, sendo as instruções recuadas à direita.
- O símbolo } deve estar alinhado ao abre-chaves correspondente.

Recuos não resolvem ambigüidades

- Exemplo:

```
if (nota >= 9)
   if (nota_anterior < nota)
      printf("Você está melhorando.");
else
   printf("Sem estudo é difícil ser aprovado.");
```

- De quem é o else acima?
  - O compilador sempre associa um else ao "if anterior mais próximo que ainda não possui um else."

- Como associar o else à instrução if (nota >= 9)?

A importância dos recuos

- Exemplo:

```
if (nota >= 9)
{
   if (nota_anterior < nota)
      printf("Você está melhorando.");
}
else
   printf("Sem estudo é difícil ser aprovado.");
```

- Neste caso, as chaves, em vez de opcionais, serão obrigatórias, pois apenas os recuos não resolvem.

Problema 6

- Dada uma temperatura em graus centígrados, apresentá-la em graus Fahrenheit. A fórmula de conversão é: F = (9 * C + 160) / 5.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  float C,F;
  printf("Digite a temperatura em graus C: ");
  scanf("%f", &C);
  F = (9 * C + 160) / 5;
  printf("Esta temperatura corresponde a %.1f graus F\n", F);
  system("PAUSE");
  return 0;
}
```

Leitura de dados

- Nos programas anteriores, os valores das variáveis eram estabelecidos em operações de atribuição. Mas agora, qual é o valor de C?

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  float C,F;
  printf("Digite a temperatura em graus C: ");
  scanf("%f", &C);
  F = (9 * C + 160) / 5;
  printf("Esta temperatura corresponde a %.1f graus F\n", F);
  system("PAUSE");
  return 0;
}
```

Leitura de dados

- Uma outra forma de atribuir valores a variáveis é a leitura de dados. Em C, usa-se a função scanf.

- Assim como printf, a função scanf pode ter vários parâmetros, sendo o primeiro uma string.

- No caso da função scanf, esta string deve conter apenas tags, separadas por espaços em branco.

- Os demais parâmetros da função scanf devem ser endereços de variáveis.

Leitura de dados

- O que acontece quando o computador executa uma instrução de leitura de dados? Exemplo:

```
scanf("%f",&C);
```

- A execução do programa é interrompida. O computador espera que o usuário digite algum valor e pressione a tecla Enter.

- Após pressionar Enter, o computador retoma a execução do programa e armazena o(s) valor(es) digitado(s) no(s) endereço(s) fornecido(s) na função scanf.

Leitura de dados

- O que difere a leitura de dados da operação de atribuição?
  - Na operação de atribuição, o valor a ser atribuído é definido antes da execução do programa, enquanto numa operação de leitura de dados, o valor atribuído é definido durante a execução.

- Em programação, diz-se que coisas são estáticas quando ocorrem antes do programa executar e dinâmicas, quando ocorrem durante a execução.

```
C = 32;
scanf("%f",&C)
```

Valor de C é estabelecido estaticamente.

Valor de C é estabelecido dinamicamente.

Leitura de dados

- Na leitura de dados, o valor digitado pelo usuário deve ser do mesmo tipo que a variável.

- Com a leitura de dados, a execução de um programa pode ser realizada para valores diferentes das variáveis.

- Porém, se o valor da variável é estabelecido de forma estática, para cada valor da variável, é necessário compilar o programa novamente.

Problema 7

- Dadas as idades (tipo int) e os pesos (tipo float) de duas pessoas, exibir quem é a pessoa mais velha e a sua idade e quem é a pessoa mais leve e o seu peso.

```
// Programa p07.c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  int idade1,idade2,maior_idade;
  int mais_velho,mais_leve;
  float peso1,peso2,menor_peso;
  printf("Digite a idade e o peso da pessoa 1: ");
  scanf("%d %f", &idade1,&peso1);
  printf("Digite a idade e o peso da pessoa 2: ");
  scanf("%d %f", &idade2,&peso2);
```

Problema 7

```
  if (idade1 > idade2)
  {
    maior_idade = idade1;
    mais_velho = 1;
  }
  else
  {
    maior_idade = idade2;
    mais_velho = 2;
  }
  if (peso1 < peso2)
  {
    menor_peso = peso1;
    mais_leve = 1;
  }
  else
  {
    menor_peso = peso2;
    mais_leve = 2;
  }
  printf("Maior idade = %d (da pessoa %d)\n",maior_idade,mais_velho);
  printf("Menor peso = %.1f (da pessoa %d)\n",menor_peso,mais_leve);
  system("PAUSE");
  return 0;
}
```