

Programação Estruturada

Estruturas condicionais

Professores Emílio Francesquini e Carla Negri Lintzmayer

2018.Q3

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC



Comandos condicionais

Comandos condicionais

Um comando condicional é aquele que permite decidir se um determinado bloco de comandos deve ou não ser executado, de acordo com o resultado de uma expressão relacional ou lógica.



- É um conjunto de instruções agrupadas
- Em C, é limitado pelos caracteres { e }

```
1  #include <stdio.h>
2
3  int main() { /* início do bloco de comandos */
4      int a;
5      a = 1;
6      return 0;
7  } /* fim do bloco de comandos */
```

Comandos condicionais

O principal comando condicional da linguagem C é o **if**, cuja sintaxe é

```
1 if (expressão relacional ou lógica)
2     um único comando;
```

ou

```
1 if (expressão relacional ou lógica) {
2     sequência de comandos;
3 }
```

Os comandos são executados somente se a expressão relacional/lógica for verdadeira.

Comandos condicionais

O programa abaixo determina se um valor é ímpar.

```
1  #include <stdio.h>
2
3  int main() {
4      int a;
5
6      scanf("%d", &a);
7      if ((a % 2) != 0) {
8          printf("O valor digitado é ímpar.\n");
9      }
10
11     return 0;
12 }
```

Comandos condicionais

Lembrando como C representa os valores falso e verdadeiro, o programa anterior é equivalente ao seguinte.

```
1  #include <stdio.h>
2
3  int main() {
4      int a;
5
6      scanf("%d", &a);
7      if (a % 2) {
8          printf ("O valor digitado é ímpar.\n");
9      }
10
11     return 0;
12 }
```

Uma variação do comando **if** é o **if/else**, cuja sintaxe é

```
1  if (expressão relacional ou lógica) {  
2      comandos executados se a expressão é verdadeira;  
3  } else {  
4      comandos executados se a expressão é falsa;  
5  }
```

Comandos condicionais

O programa a seguir determina o menor dentre dois números.

```
1  #include <stdio.h>
2  int main() {
3      int a, b;
4
5      scanf("%d", &a);
6      scanf("%d", &b);
7
8      if (a < b) {
9          printf("O menor número é: %d\n", a);
10     } else {
11         printf("O menor número é: %d\n", b);
12     }
13
14     return 0;
15 }
```

Comandos condicionais

Note que o **if** é um comando e, como tal, pode aparecer dentro do bloco de comandos de outro **if**.

Exemplo: usando **apenas** operadores relacionais e aritméticos, vamos escrever um programa que lê um número e verifica em qual dos seguintes casos o número se enquadra:

- Par e menor que 100
- Par e maior ou igual a 100
- Ímpar e menor que 100
- Ímpar e maior ou igual a 100

Comandos condicionais

```
1  #include <stdio.h>
2  int main() {
3      int a;
4
5      scanf("%d", &a);
6
7      if (a % 2 == 0) {
8          if (a < 100)
9              printf("0 número é par e menor que 100\n");
10             else
11                 printf("0 número é par e maior ou igual a 100\n");
12         } else {
13             if (a < 100)
14                 printf("0 número é ímpar e menor que 100\n");
15             else
16                 printf("0 número é ímpar e maior que 100\n");
17         }
18
19         return 0;
20     }
```

Comandos condicionais

```
1  #include <stdio.h>
2
3  int main() {
4      int a;
5      scanf("%d", &a);
6
7      if ((a % 2 == 0) && (a < 100))
8          printf("0 número é par e menor que 100\n");
9      if ((a % 2 == 0) && (a >= 100))
10         printf("0 número é par e maior ou igual a 100\n");
11     if ((a % 2 != 0) && (a < 100))
12         printf("0 número é ímpar e menor que 100\n");
13     if ((a % 2 != 0) && (a >= 100))
14         printf("0 número é ímpar e maior que 100\n");
15
16     return 0;
17 }
```

Comandos condicionais

```
1  if (cond1) {  
2      if (cond2)  
3          comando1;  
4  } else  
5      comando2;
```

Quando o **comando2** é executado?

Comandos condicionais

```
1  if (cond1) {  
2      if (cond2)  
3          comando1;  
4      else  
5          comando2;  
6  } else {  
7      if (cond3)  
8          comando3;  
9      else  
10         comando4;  
11 }
```

Quando o **comando4** é executado?

Comandos condicionais

Use chaves e indentação para deixar claro a qual comando condicional um outro comando pertence!!

```
1  if (cond1)
2  if (cond2)
3      comando1;
4  else
5      comando2;
```

Quando o **comando2** é executado?

Usando chaves e indentação no exemplo anterior para deixar mais claro:

```
1  if (cond1) {  
2      if (cond2)  
3          comando1;  
4      else  
5          comando2;  
6  }
```

Comandos condicionais

```
1  #include <stdio.h>
2  int main() {
3      int a;
4      scanf("%d", &a);
5
6      if (a > 3) {
7          if (a < 7)
8              printf("a\n");
9      } else {
10         if (a > -10)
11             printf("b\n");
12         else
13             printf("c\n");
14     }
15
16     return 0;
17 }
```

O que será impresso se digitarmos:

- 5
- -12
- 9

Mais sobre o comando de atribuição

O comando de atribuição em C é `=`.

Em C, uma expressão de atribuição tem valor igual ao valor da variável à esquerda.

```
1  #include <stdio.h>
2
3  int main() {
4      int a, b;
5      printf("%d\n", (a = 4));
6      printf("%d\n", (a = 0));
7      printf("%d\n", (a = 4+5));
8      printf("%d\n", (a = b = 4));
9      return 0;
10 }
```

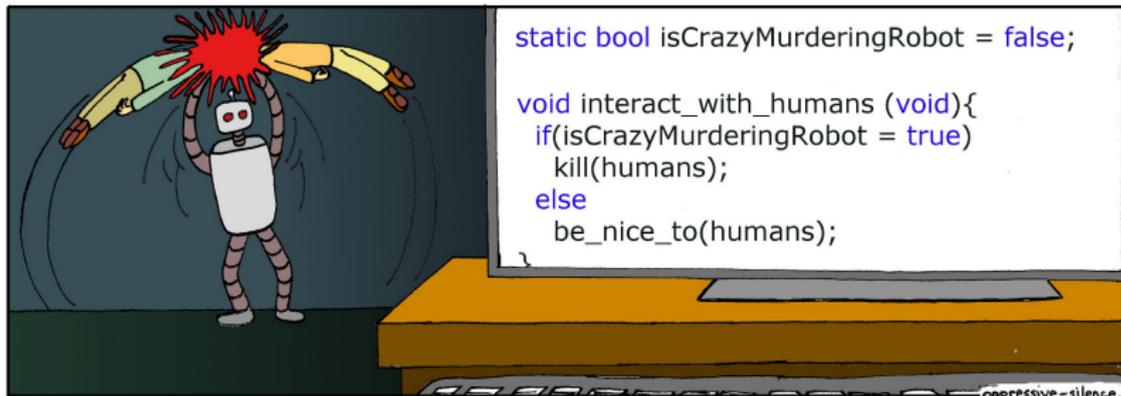
Comandos condicionais

Não confunda o comando de atribuição com o teste de igualdade (`==`), pois isto pode gerar erros!

```
1  #include <stdio.h>
2
3  int main() {
4      int a = 2;
5
6      if (a = 3) {
7          printf("fazer algo se a for 3\n");
8      } else {
9          printf("fazer algo se a não for 3\n");
10     }
11
12     return 0;
13 }
```

O programa acima imprime “fazer algo se a for 3”.

Comandos condicionais



Exercícios

Exercícios

A solução abaixo está correta para classificar um número como par e menor que 100, ou par e maior ou igual a 100, etc., como no exemplo visto anteriormente?

```
1  #include <stdio.h>
2  int main() {
3      int a;
4      scanf("%d", &a);
5
6      if ((a % 2 == 0) && (a < 100))
7          printf("0 número é par e menor que 100\n");
8      else if (a >= 100)
9          printf("0 número é par e maior ou igual a 100\n");
10     if ((a % 2 != 0) && (a < 100))
11         printf("0 número é ímpar e menor que 100\n");
12     else if (a >= 100)
13         printf("0 número é ímpar e maior que 100\n");
14
15     return 0;
16 }
```

Escreva um programa que lê um número inteiro do teclado e imprime "SIM" se o número for par e maior do que 10 ou se for ímpar e menor do que 50. Caso contrário o programa deve imprimir "NAO".

Escreva um programa lê três números e imprime o maior deles.

Escreva um programa lê três números e os imprime em ordem crescente.

Mais sobre comandos condicionais

Comandos condicionais if-else

- Vamos fazer um programa que calcula a área de três tipos de objetos geométricos: quadrado, retângulo e círculo.
- Primeiramente deve ser lido um caractere que indica o tipo de objeto a ter a área calculada: 'q' para quadrado, 'r' para retângulo e 'c' para círculo.
- Em seguida deverão ser lidas as dimensões do objeto:
 - Para um quadrado deve ser lido o tamanho de um lado.
 - Para um retângulo devem ser lidos os tamanhos de cada lado.
 - Para um círculo deve ser lido o raio.
- Em seguida o programa faz o cálculo da área do objeto e a imprime.
- Se o usuário digitar um caractere diferente de 'q', 'r', e 'c' o programa deverá imprimir uma mensagem de erro.

Comandos condicionais if-else

```
1  int main() {
2      char op;
3      double lado_quad, lado1_ret, lado2_ret, raio;
4      scanf("%c", &op); /* lendo opção (q, r ou c) */
5      if (op == 'q') {
6          ...
7      }
8      if (op == 'r') {
9          ...
10     }
11     if (op == 'c') {
12         ...
13     }
14     if (op != 'q' && op != 'r' && op != 'c') {
15         printf("Opção inválida!\n");
16     }
17     return 0;
18 }
```

Comandos condicionais if-else

```
1  int main() {
2      char op;
3      double lado_quad, lado1_ret, lado2_ret, raio;
4      scanf("%c", &op); /* lendo opção (q, r ou c) */
5      if (op == 'q') {
6          scanf("%lf", &lado);
7          printf("A área é: %.2f\n", lado_quad * lado_quad);
8      }
9      if (op == 'r') {
10         scanf("%lf %lf", &lado1_ret, &lado2_ret);
11         printf("A área é: %.2f\n", lado1_ret * lado2_ret);
12     }
13     if (op == 'c') {
14         scanf("%lf", &raio);
15         printf("A área é: %.2f\n", 3.1415 * raio * raio);
16     }
17     if (op != 'q' && op != 'r' && op != 'c') {
18         printf("Opção inválida!\n");
19     }
20     return 0;
21 }
```

Comandos condicionais if-else

```
1  int main() {
2      char op;
3      double lado_quad, lado1_ret, lado2_ret, raio;
4      scanf("%c", &op); /* lendo opção (q, r ou c) */
5      if (op == 'q') {
6          scanf("%lf", &lado_quad);
7          printf("A área é: %.2f\n", lado_quad * lado_quad);
8      } else {
9          if (op == 'r') {
10             scanf("%lf %lf", &lado1_ret, &lado2_ret);
11             printf("A área é: %.2f\n", lado1_ret * lado2_ret);
12         } else {
13             if (op == 'c') {
14                 scanf("%lf", &raio);
15                 printf("A área é: %.2f\n", 3.1415 * raio * raio);
16             } else {
17                 printf("Opção inválida!\n");
18             }
19         }
20     }
21     return 0;
22 }
```

Comandos condicionais if-else

- Na nova versão do programa, assim que um **if** for verdadeiro, nenhum dos demais **ifs** posteriores serão verificados, pois estes estão dentro do caso **else** do **if** verdadeiro.
- Perceba também que só será impresso *'Opção inválida'* quando as condições dos três **ifs** anteriores forem falsas.
- É muito comum este tipo de construção em programas, onde só deve ser executada uma opção dentre todas as alternativas possíveis.
 - No programa anterior este é o caso, pois temos os casos de cálculo da área e o caso de opção inválida.
 - Sabemos que somente um deles deverá ser executado para qualquer caractere inicial lido.

Comandos if-else-if encaixados

- Uma coisa muito comum em programação é o teste de várias alternativas exclusivas.
- Suponha a busca por informações de um determinado aluno, onde temos apenas o seu RA.
- Podemos usar uma construção simples com **ifs**, como no exemplo anterior.

Comandos if-else-if encaixados

```
1  ...
2  printf("Digite RA do aluno: ");
3  scanf("%d", &ra);
4
5  if (ra == 10129) {
6      printf("Maria Cândida Moreira Telles\n");
7  }
8  if (ra == 33860) {
9      printf("Larissa Garcia Alfonsi\n");
10 }
11 if (ra == 33967) {
12     printf("Leonardo Kozlowiski Kenupp\n");
13 }
14 if (...)
15 ...
```

- Porém todos os testes condicionais serão executados!
- Quando apenas uma de várias alternativas é verdadeira, podemos usar a construção **if-else-if**.

Comandos if-else-if encaixados

```
1  ...
2  printf("Digite RA do aluno:");
3  scanf("%d", &ra);
4  if (ra == 10129) {
5      printf("Maria Cândida Moreira Telles\n");
6  } else {
7      if (ra == 33860) {
8          printf("Larissa Garcia Alfonsi\n");
9      } else {
10         if (ra == 33967) {
11             printf("Leonardo Kozlowiski Kenupp\n");
12         } else {
13             ...
14         }
15     }
16 }
17 ...
```

Comandos if-else-if encaixados

Podemos ainda reestruturar o código da seguinte forma, uma vez que cada **else** possui apenas um comando dentro dele (um **if-else**).

```
1  ...
2  printf("Digite RA do aluno:");
3  scanf("%d", &ra);
4  if (ra == 10129) {
5      printf("Maria Cândida Moreira Telles\n");
6  } else if (ra == 33860) {
7      printf("Larissa Garcia Alfonsi\n");
8  } else if (ra == 33967) {
9      printf("Leonardo Kozlowiski Kenupp\n");
10 } else if (...) {
11     ...
12 }
13 ...
```

- Na construção **if-else-if**, quando uma condição é verdadeira, o bloco de comandos correspondente será executado.
- Após a execução do bloco de comandos, as outras alternativas não serão testadas.
- O último **else** pode ser utilizado como uma opção padrão quando nenhuma das condições dos **ifs** anteriores for verdadeira.

Comandos if-else-if encaixados

Considere novamente o programa que calcula a área de objetos, mas com a construção **if-else-if** e simplificação de chaves.

```
1  int main() {
2      char op;
3      double lado_quad, lado1_ret, lado2_ret, raio;
4      scanf("%c", &op); /* lendo opção (q, r ou c) */
5      if (op == 'q') {
6          scanf("%lf", &lado_quad);
7          printf("A área é: %.2f\n", lado_quad * lado_quad);
8      } else if (op == 'r') {
9          scanf("%lf %lf", &lado1_ret, &lado2_ret);
10         printf("A área é: %.2f\n", lado1_ret * lado2_ret);
11     } else if (op == 'c') {
12         scanf("%lf", &raio);
13         printf("A área é: %.2f\n", 3.1415 * raio * raio);
14     } else {
15         printf("Opção inválida!\n");
16     }
17     return 0;
18 }
```

Comandos if-else-if encaixados

- No brasileirão, 20 times disputam o título em dois turnos.
- No primeiro turno, os times jogam entre si uma única vez.
- Os jogos do segundo turno ocorrem na mesma ordem que no primeiro, apenas invertendo-se o mando de campo.
- Os times são classificados por pontos.
- Caso dois times atinjam o mesmo número de pontos, eles são desempatados aplicando-se os seguintes critérios, em ordem:
 1. número de vitórias (maior melhor)
 2. saldo de gols (maior melhor)
 3. gols marcados (maior melhor)
 4. número de cartões vermelhos (menor melhor)
 5. número de cartões amarelos (menor melhor)
- Faça um programa que leia as cinco informações acima de dois times e decida qual time vence o desempate.

Comandos if-else-if encaixados

Começamos lendo os dados requeridos.

```
1  int main() {
2      int vitorias1, saldo1, gols1, vermelho1, amarelo1;
3      int vitorias2, saldo2, gols2, vermelho2, amarelo2;
4
5      printf("Lendo dados do time 1\n");
6      scanf("%d", &vitorias1);
7      scanf("%d", &saldo1);
8      scanf("%d", &gols1);
9      scanf("%d", &vermelho1);
10     scanf("%d", &amarelo1);
11
12     printf("Lendo dados do time 2\n");
13     scanf("%d", &vitorias2);
14     scanf("%d", &saldo2);
15     scanf("%d", &gols2);
16     scanf("%d", &vermelho2);
17     scanf("%d", &amarelo2);
18     ...
```

Comandos if-else-if encaixados

Agora testamos quem possui mais vitórias, para decidir o vencedor.

```
1 ...
2 if (vitorias1 > vitorias2)
3     printf("Time 1 ganha do Time 2\n");
4 else if (vitorias1 < vitorias2)
5     printf("Time 2 ganha do Time 1\n");
```

O que podemos deduzir se as duas condições dos **ifs** acima forem falsas?

Comandos if-else-if encaixados

```
1  if (vitorias1 > vitorias2)
2      printf("Time 1 ganha do Time 2\n");
3  else if (vitorias1 < vitorias2)
4      printf("Time 2 ganha do Time 1\n");
5  else if (saldo1 > saldo2)
6      printf("Time 1 ganha do Time 2\n");
7  else if (saldo1 < saldo2)
8      printf("Time 2 ganha do Time 1\n");
9  else if (gols1 > gols2)
10     printf("Time 1 ganha do Time 2\n");
11  else if (gols1 < gols2)
12     printf("Time 2 ganha do Time 1\n");
13  else if (vermelho1 < vermelho2)
14     printf("Time 1 ganha do Time 2\n");
15  else if (vermelho1 > vermelho2)
16     printf("Time 2 ganha do Time 1\n");
17  else if (amarelo1 < amarelo2)
18     printf("Time 1 ganha do Time 2\n");
19  else if (amarelo1 > amarelo2)
20     printf("Time 2 ganha do Time 1\n");
```

É possível que todas as condições avaliadas nesse código sejam falsas?

Comandos if-else-if encaixados

```
1  ...
2  else if (vermelho1 < vermelho2)
3      printf("Time 1 ganha do Time 2\n");
4  else if (vermelho1 > vermelho2)
5      printf("Time 2 ganha do Time 1\n");
6  else if (amarelo1 < amarelo2)
7      printf("Time 1 ganha do Time 2\n");
8  else if (amarelo1 > amarelo2)
9      printf("Time 2 ganha do Time 1\n");
10 else
11     printf("Times continuam empatados!\n");
```

Pela regra do campeonato, se os times continuarem empatados então o desempate se dará por sorteio!

Informações extras: o comando
switch

O comando `switch`

O objetivo do comando **switch** é simplificar uma construção **if-else-if** encaixados quando as condições ocorrem sobre uma variável **inteira** ou **caractere**.

```
1 switch (variável) {
2     case valor1:
3         /* comandos */
4         break;
5     case valor2:
6         /* comandos */
7         break;
8     ...
9 }
```

O comando switch

```
1 printf("Digite o RA: ");
2 scanf("%d", &ra);
3
4 switch (ra) {
5     case 10129:
6         printf("Maria Cândida Moreira Telles\n");
7         break;
8     case 33860:
9         printf("Larissa Garcia Alfonsi\n");
10        break;
11    case 33967:
12        printf("Leonardo Kozlowiski Kenupp\n");
13        break;
14 }
```

O comando `switch`

- Os comandos começam a ser executados a partir do ponto onde o valor da variável corresponde ao valor entre “`case`” e “`:`”
- São executados todos os comandos até que se encontre um comando `break` ou até que se chegue ao final do bloco de comandos do `switch`

O comando `switch`: valor padrão

Você pode utilizar uma condição `default`. A execução dentro dela ocorre se nenhuma outra condição for verdadeira (assim como o último `else` do `if-else-if` encaixados).

```
1 switch (variável) {
2     case valor1:
3         /* comandos */
4         break;
5     case valor2:
6         /* comandos */
7         break;
8     ...
9     default:
10        /* comandos */
11 }
```

O comando switch: valor padrão

```
1 printf("Digite o RA: ");
2 scanf("%d", &ra);
3
4 switch (ra) {
5     case 10129:
6         printf("Maria Cândida Moreira Telles\n");
7         break;
8     case 33860:
9         printf("Larissa Garcia Alfonsi\n");
10        break;
11    default:
12        printf("O aluno não está matriculado\n");
13 }
```

Exercícios

Quando ações são vendidas ou compradas por meio de um corretor, a comissão do corretor é muitas vezes calculada usando uma escala que depende do valor das ações negociadas. Escreva um programa que calcule o valor da comissão a partir do valor da transação informado pelo usuário, sabendo-se que o corretor cobra os valores indicados abaixo e que a **comissão mínima é de R\$ 39,00**:

- Até R\$ 2.500,00, comissão de R\$30+1,7%
- R\$2.500,01 até R\$6.250,00, comissão de R\$56 + 0,66%
- R\$6.250,01 até R\$20.000,00, comissão de R\$76 + 0,34%
- R\$20.000,01 até R\$50.000,00, comissão de R\$100 + 0,22%
- R\$50.000,01 até R\$500.000,00, comissão de R\$155 + 0,11%
- Mais que R\$ 500.000,00, comissão de R\$255 + 0,09%

Exercícios

Escreva um programa que, dadas as notas das duas provas (P_1 e P_2), da média ponderada das notas dos laboratórios (L) e do projeto (T) de um aluno de PE, calcula o conceito final desse aluno antes da recuperação.

Recapitulando, a média final MF é dada por

$$MF = \begin{cases} \min\{P_1 + P_2, L, T\} & \text{se } P_1 + P_2 < 10 \text{ ou } L < 5 \text{ ou } T < 5 \\ 0.2P_1 + 0.3P_2 + 0.3L + 0.2T & \text{caso contrário} \end{cases}$$

E o conceito final é

- A se $MF \geq 8.5$
- B se $7.0 \leq MF < 8.5$
- C se $6.0 \leq MF < 7.0$
- D se $5.0 \leq MF < 6.0$
- F se $0.0 \leq MF < 5.0$

A primeira linha da entrada contém 4 números reais positivos P_1 , P_2 , L e T que armazenam as notas das provas, das listas e do projeto, respectivamente. Considere que cada número é maior ou igual a 0 e menor ou igual a 10 .

O programa deve imprimir o texto "Conceito final = X", com X devidamente substituído pelo conceito final correspondente às notas dadas.

Faça um programa que leia uma temperatura em Celsius (resp. em Fahrenheit) e devolva a temperatura correspondente em Fahrenheit (resp. em Celsius).

A primeira linha da entrada contém um caractere (que pode ser F ou C) e a segunda contém um número real, que corresponde a uma temperatura (em Fahrenheit ou Celsius, dependendo do caractere lido).

O programa deve imprimir um único número real em uma linha, com 3 casas decimais, que corresponde à temperatura lida na outra unidade de medida.

Construa uma calculadora que efetue as 4 operações aritméticas básicas além do resto da divisão.

A primeira linha da entrada contém um valor inteiro x qualquer. A segunda linha contém um caractere que indica o operador (que pode ser $+$ para soma, $-$ para subtração, $*$ para multiplicação, $/$ para divisão ou $\%$ para resto de divisão). A terceira linha contém um outro valor inteiro y qualquer.

O programa deve imprimir em uma linha um único número inteiro, resultado da operação dada na entrada, ou então a mensagem “erro” se não for possível realizar a conta.

Obs.: para a entrada “5 / 3” a saída do programa deve ser “1”.

Escreva um programa que receba três valores e decida, caso eles formem um triângulo, se o mesmo é retângulo, isósceles, equilátero ou escaleno. Seu programa deve lidar com a possibilidade de os números lidos serem negativos.

A entrada consiste de três números reais.

Seu programa deve imprimir uma ou mais linhas, cada uma contendo apenas uma das cinco opções a seguir, de acordo com os valores lidos: “retângulo”, “isósceles”, “equilátero”, “escaleno” ou “os valores não formam um triângulo”.

Note que um triângulo pode ter mais de uma classificação.

Faça um programa que leia um ano e imprima se ele é bissexto ou não.

Todos os anos múltiplos de 400 são bissextos. Não sendo múltiplo de 400, são bissextos todos os anos múltiplos de 4 mas que não são múltiplos de 100.

A entrada consiste de um único número inteiro, que representa o ano.

Seu programa deve imprimir, em uma linha, a palavra “Sim ou “Não”, indicando se o ano lido é bissexto ou não.

Escreva um programa que determina a data cronologicamente maior de duas datas fornecidas pelo usuário.

A entrada consiste de 6 valores inteiros, que representam, respectivamente, o dia, mês e ano da primeira data seguidos do dia, mês e ano da segunda data.

Seu programa deve imprimir uma única linha com o texto “dd/mm/aaaa”, indicando corretamente a maior das duas datas.