

Execução sequencial e condicional

Programação I

2022/23

Salvador Abreu

spa@uevora.pt

Teresa Gonçalves

tcg@uevora.pt

Departamento de Informática, ECT-UÉ

Sumário

Revisão

Execução condicional

Execução alternativa

Condicionais encadeados

Condicionais encaixados



Revisão

Tipos

Inteiro: int

Pode ser positivo ou negativo

Real: float

Melhor aproximação aos números reais

Caráter: char

Um carácter



Expressões lógicas

Valores

0: falso

$\neq 0$: verdade

Operadores

&& : e

|| : ou

! : não

Avaliação mínima ou “short circuit” para && e ||

Precedência

! > && > ||

Operadores relacionais

== != < <= > >=



Avaliação mínima («curto-circuito»)

Inutilidade de avaliação de todos os operandos

$x \parallel y$

Se x é falso então y , senão x

$x \&\& y$

Se x é falso então x , senão y

$!x$

Se x é falso então verdade, senão falso



Utilização de expressões lógicas

Comparações ou testes

Igualdade, desigualdade

Representação de um conjunto de características

Através de conjunções (e), disjunções (ou) e negações

Exemplos

Verificar se x é potência 2 de y

Verificar se x é divisível por 5 e múltiplo de 3

Exemplos

x é o quadrado de y ?

`x == y * y`

z é divisível por 5 e múltiplo de 3?

`(z%5 == 0) && (z%3 == 0)`

`z%5 == 0 && z%3 == 0`

`!(z%5) && !(z%3)`

`!(z%5 || z%3)`



Conversão de tipos (explícita)

(float) expr_numerica

converte expr para um valor real

(int) expr_numerica

maior inteiro menor que expr (parte inteira do número)

(char) expr_inteira

Carácter correspondente ao código ASCII do inteiro (8 bits)

(int) carácter

Código ASCII correspondente ao carácter



Execução sequencial

Execução sequencial

Uma instrução pode ser

- Uma declaração

```
int xpto = 10, xpti;
```

O ";" faz parte da instrução

- Uma expressão (com efeitos secundários)

```
xpto = xpto * 10;  
printf ("Bom dia!\n");
```

- Um grupo de instruções (instrução composta ou "bloco")

```
{  
    INSTR1  
    INSTR2  
    ...  
}
```

A instrução "{...}" não termina com ";"

Execução condicional

Execução condicional

Altera o comportamento do programa de acordo com determinadas condições

- **Instrução if**

```
if (CONDIÇÃO)  
    <instrução quando a condição é verdadeira>
```

- **Se quisermos que a instrução seja um bloco:**

```
if (CONDIÇÃO) {  
    <instruções quando a condição é verdadeira>  
}
```



Exercícios

Somar 1 se o número x for par

Indicar se é m é mês de férias (7, 8, 9)

Indicar se uma pessoa p tem peso normal ($18.5 \leq \text{IMC} < 25$)

$\text{IMC} = \text{peso} \div \text{altura}^2$ (peso em kg, altura em metros)



Execução alternativa

Especifica comportamentos diferentes caso a condição seja verdadeira ou falsa

- **Instrução if-else**

```
if (CONDIÇÃO)  
    <instrução quando a condição é verdadeira>  
else  
    <instrução quando a condição é falsa>
```

Exercício

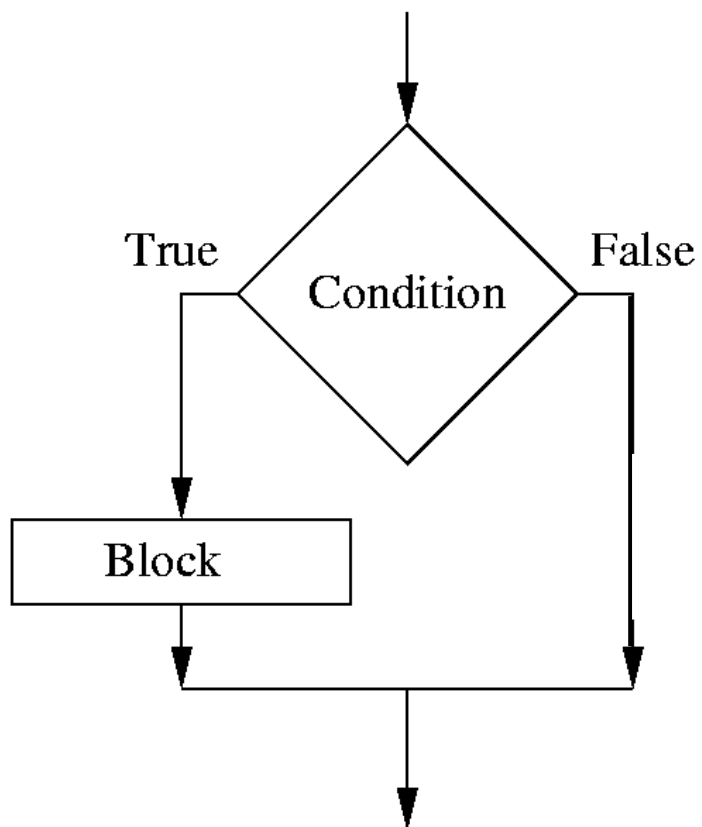
Indicar se um inteiro x é par ou ímpar

Exercício: par ou ímpar

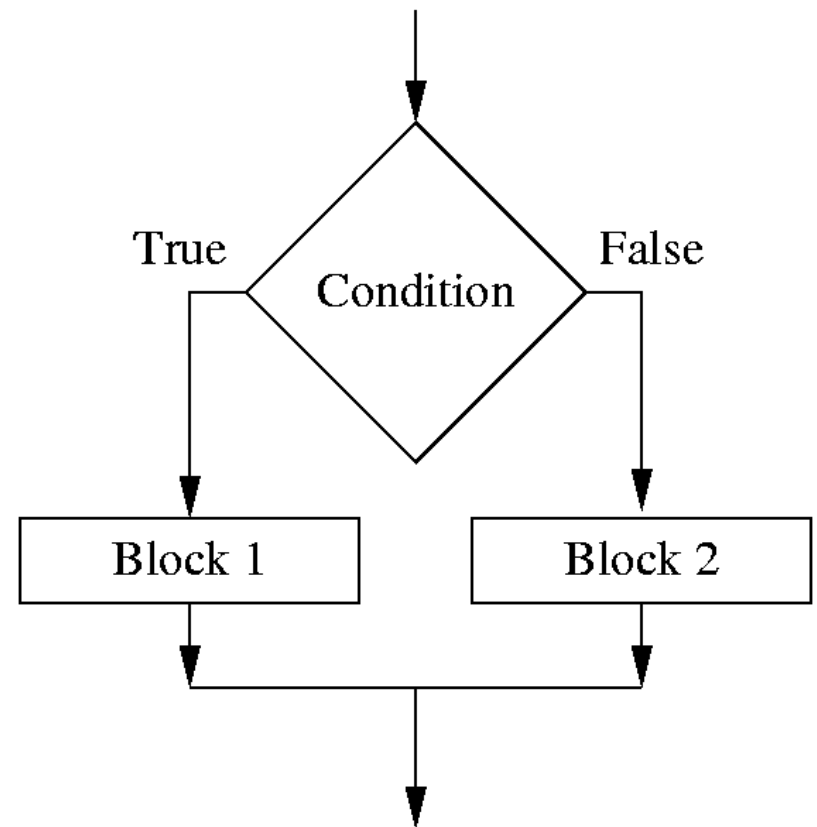
```
if (x%2 == 0) {  
    printf( "é par");  
}  
else {  
    printf( "é impar");  
}
```


Fluxogramas

Condicional: if



Alternativa: if-else



Alternativas encadeadas

Instrução **if - else if**

if (COND1)

<fazer quando a COND1 é verdadeira>

else if (COND2)

<fazer quando a COND1 é falsa e COND2 é verdadeira>

...

else

<fazer quando todas as condições anteriores são falsas>

Alternativas encadeadas (instruções compostas)

Instrução **if - else if**

```
if (COND1) {  
    <fazer quando a COND1 é verdadeira>  
}
```

```
else if (COND2) {  
    <fazer quando COND1 é falsa e COND2 é verdadeira>  
}
```

...

```
else {  
    <fazer quando todas as condições anteriores são falsas>  
}
```

Características

Apenas é executado um dos ramos

Podem existir vários **else if**

Pode não existir um **else**

Apenas as instruções referentes à 1ª condição verificada como verdadeira são executadas

... mesmo existindo outras condições verdadeiras

Exercícios

Indicar se x é maior, menor ou igual a y

Indicar o nº de dias do mês m

x é maior, menor ou igual a y

```
if (x < y)
    printf("x é menor que y");
else if (x > y)
    printf("x é maior que y");
else
    printf("x é igual a y");
```

Nº dias do mês m

```
if (m==2)
    printf("tem 28 ou 29 dias");
else if (m==4 || m==6 || m==9 || m==11)
    printf("tem 30 dias");
else
    printf("tem 31 dias");
```

Condicionais encaixados

```
if (x < y)
    printf("x é menor que y");
else {
    if (x > y)
        printf("x é maior que y");
    else
        printf("x é igual a y");
}
```

Podem tornar-se difíceis de ler

Apesar da indentação facilitar a compreensão

Exemplo

Por vezes são usados indevidamente

```
if (x > 0)
    if (x < 10)
        printf ("%d é inteiro positivo com um algarismo\n", x);
```

Qual a forma mais “correta”?

Exemplo

```
if (COND1) {  
    xpto = foo();  
    x = 1;  
    if (COND2)  
        faz_isto();  
    else {  
        faz_aquilo();  
        x = 2;  
    }  
}
```

