

Iteração

Programação I
2021.2022

Teresa Gonçalves
tcg@uevora.pt

Departamento de Informática, ECT-UÉ

Sumário

Revisão

Iteração

Controlo de fluxo

Revisão

Execução condicional

Instrução if

```
if( <condição> )  
    <bloco de instruções quando a condição é verdadeira>;
```

Bloco de instruções:

- Uma instrução
- Um bloco de instruções delimitado pelo par { }

Instrução if-else

```
if( <condição> )  
    <bloco de instruções quando a condição é verdadeira>;  
else  
    <bloco de instruções quando a condição é falsa>;
```

Execução condicional

Instrução if-else if

```
if( <condição1> )
```

```
    <bloco de instruções quando condição1 é verdadeira>
```

```
else if( <condição2> )
```

```
    <bloco de instruções quando condição1 é falsa e  
    condição2 é verdadeira>
```

```
...
```

```
else
```

```
    <bloco de instruções quando todas as condições  
    anteriores são falsas>
```

Características

Pode não existir um else

Podem existir inúmeros else if

Apenas é executado um dos ramos

Apenas as instruções referentes à 1ª condição verdadeira são executadas

Condicionais encadeados vs. encaixados

Encadeado

Utilização da instrução `else if`

Encaixado

Instrução `if` como corpo do `else`

Condicionais encaixados

Podem tornar a leitura/compreensão mais difícil

Podem ser utilizados indevidamente

Iteração

Problema - *countdown*

Mostrar uma contagem decrescente: de 5 a 0

Solução 1

```
printf( "%d\n", 5 );  
printf( "%d\n", 4 );  
printf( "%d\n", 3 );  
printf( "%d\n", 2 );  
printf( "%d\n", 1 );  
printf( "%d\n", 0 );
```

Problema - *countdown*

E se a contagem começar em 20?

E se a contagem começar em N?

O problema é idêntico!

Varia o **número de vezes** que a instrução `printf` é invocada e o **valor** que é mostrado

Iteração

Capacidade de executar um bloco de instruções **várias vezes!**

Instrução while

```
while( <condicao> )  
    <bloco de instruções>
```

Bloco de instruções

- Uma instrução
- Um bloco de instruções delimitado pelo par { }

Fluxo de execução

1. Avalia a condição, obtendo o valor Verdade ou Falso
 - a. Se Falso, sai da instrução while e continua com próxima instrução
 - b. Se Verdade, executa o bloco de instruções e volta ao ponto 1

Problema *countdown* (2)

```
int n;  
n=5;  
while( n>=0 ){  
    printf( "%d\n", n );  
    n = n-1;  
}
```

Condição while

Para o ciclo terminar...

... o valor das variáveis da condição devem ser alteradas no corpo do while!

Nem sempre é fácil verificar a convergência

```
while( n>1 ){  
    printf( "%d\n", n );  
    if( n%2 == 0 )  
        n = n/2;  
    else  
        n = n*3+1;  
}
```

Qual a sequência quando n é 5?

E quando n é 7?

Exemplos

Calcular a média de 5 números

Calcular a média até à introdução de um valor negativo

Calcular a raíz quadrada de um número

Calcular a média de 5 números

```
int n;  
float num, soma, media;  
soma=0;  
n=1;  
while( n<=5 ){  
    printf( "Introduza um numero: " );  
    scanf( "%d", &num );  
    soma=soma+num;  
    n=n+1;  
}  
media = soma/5;  
printf( "A média dos números introduzidos é %f.\n", media );
```

Calcular a média até à introdução de um valor negativo (1)

```
int contador;  
float num, soma, media;  
soma=0;  
contador=0;  
printf("Introduza um numero (<0 para terminar): ");  
scanf( "%d", &num );  
while( num>=0 ){  
    soma=soma+num;  
    contador=contador+1;  
    printf("Introduza um numero (<0 para terminar):");  
    scanf( "%d", &num );  
}  
if( contador>0 ){  
    media=soma/contador;  
    printf("A media dos numeros introduzidos é %f.", media);  
} else  
    printf("Não foi introduzido nenhum número");
```


Calcular a média até à introdução de um valor negativo (2)

```
int contador;
float num, soma, media;
soma=0;
contador=0;
num=0;
while( num>=0 ){
    printf("Introduza um numero (<0 para terminar): ");
    scanf( "%d", &num );
    if( num >=0 ){
        soma=soma+num;
        contador=contador+1;
    }
}
if( contador>0 ){
    media=soma/contador;
    printf("A media dos numeros introduzidos é %f.", media);
} else
    printf("Não foi introduzido nenhum número");
```

Calcular a média até à introdução de um valor negativo

```
int contador;
float num, soma, media;
soma=0;
contador=0;
printf("Introduza um numero (<0 para terminar): ");
scanf( "%d", &num );
while( num>=0 ){
    soma=soma+num;
    contador=contador+1;
    printf("Introduza um numero (<0 para
terminar):");
    scanf( "%d", &num );
}
if( contador>0 ){
    media=soma/contador;
    printf("A media dos numeros introduzidos é %f.",
media);
} else
    printf("Não foi introduzido nenhum número");
```

Porque testamos $num \geq 0$

```
int contador;
float num, soma, media;
soma=0;
contador=0;
num=0;
while( num>=0 ){
    printf("Introduza um numero (<0 para terminar):
");
    scanf( "%d", &num );
    if( num >=0 ){
        soma=soma+num;
        contador=contador+1;
    }
}
if( contador>0 ){
    media=soma/contador;
    printf("A media dos numeros introduzidos é %f.",
media);
} else
    printf("Não foi introduzido nenhum número");
```

Raíz quadrada: método de Newton

Algoritmo

Começa com uma estimativa **x** da raiz quadrada do número **a**

Calcula uma nova estimativa, **y**

Termina quando a diferença entre 2 estimativas consecutivas é desprezível

Estimativa

$$y = \frac{x + a/x}{2}$$

Calcular a raiz quadrada de um número

Variáveis

Número: a

Estimativa inicial: x

Valor de paragem: epsilon

$$y = \frac{x + a/x}{2}$$

```
int main(){  
    float a, x, y, epsilon;  
    printf("Introduza um numero: ");  
    scanf("%f", &a);  
    epsilon = 0.00001;  
    x = a/2;          /* estimativa anterior */  
    y = (x+a/x)/2;    /* nova estimativa */  
    while( (y-x)*(y-x) >= epsilon*epsilon ){  
        x = y;  
        y = (x+a/x)/2;  
    }  
    printf( "A raiz quadrada é %f.\n", y );  
}
```

Instrução do-while

```
do  
    <bloco de instruções>  
while( <condicao> )
```

Bloco de instruções

- Uma instrução
- Um bloco de instruções delimitado pelo par { }

Fluxo de execução

1. Executa o bloco de instruções
2. Avalia a condição, obtendo o valor Verdade ou Falso
 - a. Se Falso, sai da instrução while e continua com próxima instrução
 - b. Se Verdade volta ao ponto 1

Iteração

```
while( condição )  
    bloco_instruções
```

```
do  
    bloco_instruções  
while( condição );
```

Controlo de fluxo

Transferência de controlo de fluxo

Execução

Normalmente as instruções são executadas de forma sequencial

As estruturas de controlo permitem definir diferentes possíveis sequências

Estruturas de controlo

Condicional

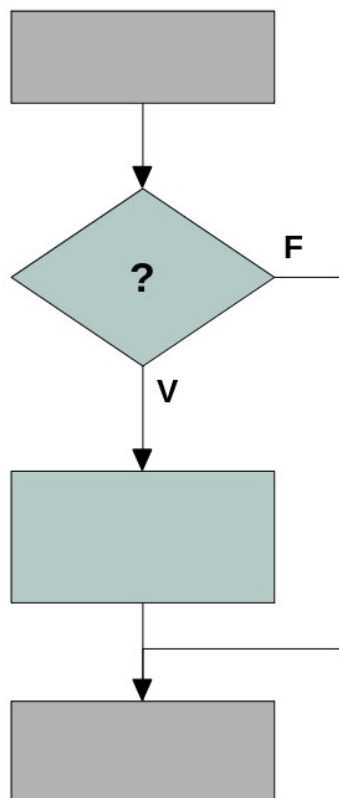
if, if-else, if-else if

Iteração

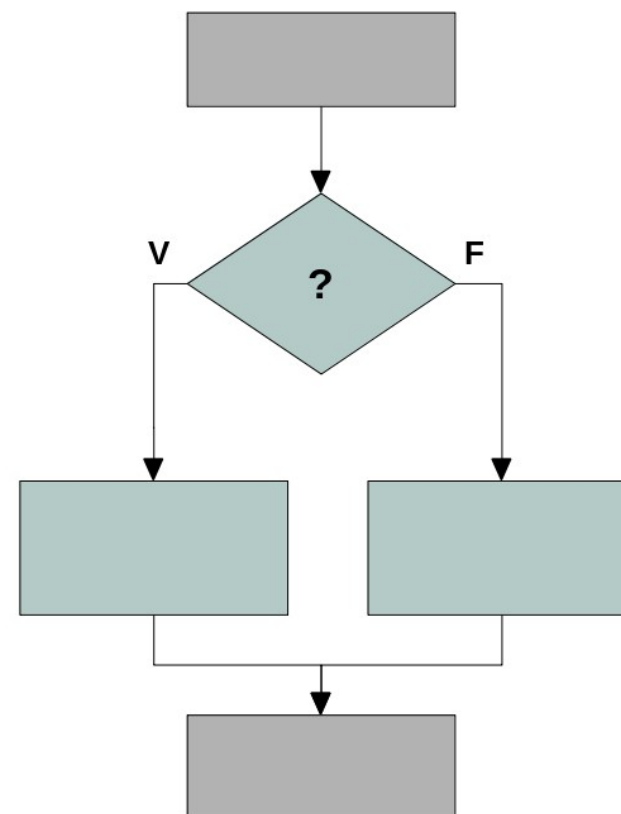
while, do-while

Condicional

if

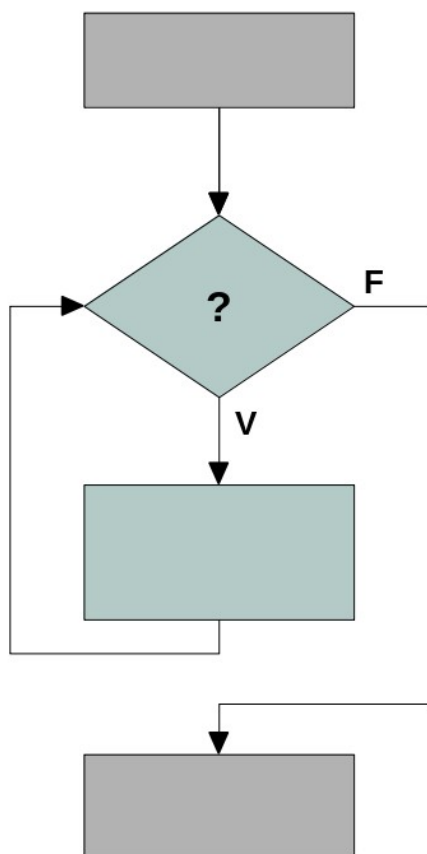


if-else



Iteração

while



do-while

