Algoritmia e

Programação

Controlo de fluxo: estruturas de repetição



Repetir uma acção

* Muitas vezes, para resolver um problema temos que repetir um conjunto de acções até que uma condição se verifique.
* Por exemplo, ir até à praia de bicicleta…
  + “Tenho que pedalar até chegar à praia.”



**APROG - LETI**

Repetir uma acção

* Muitas vezes, para resolver um problema temos que **repetir** um conjunto de acções num **ciclo**, até queuma condição se verifique.
* Por exemplo, ir até à praia de bicicleta…
* “Tenho que pedalar até chegar à praia.”



Acção repetida Condição



**APROG - LETI**

Insistir antes de prosseguir!

•

•

Um algoritmo deverá permitir exprimir que acções deverão ser **repetidas** consoante uma **condição**.

A avaliação da condição só pode resultar em dois valores possíveis:

* + **VERDADEIRO**
  + **FALSO**
* **Quando não há condições para repetir, o algoritmo deverá prosseguir com as acções seguintes!**



**APROG - LETI**

Ciclos controlados por uma condição



**APROG - LETI**

Ciclos controlados por uma condição

•

•

A execução do bloco de acções de um ciclo são controladas através do valor lógico de uma condição.

Distinguem-se dois tipos de ciclo:

* O ciclo que permite que o bloco de acções seja executado **incondicionalmente** pela **primeira vez**.
  + Executa o ciclo pelo menos uma vez.
* O ciclo em que o bloco de acções é executado **sempre dependendo da condição**.
  + Pode nunca executar o ciclo!



**APROG - LETI**

REPETIR … ENQUANTO

* Esta construção é utilizada nos casos em que o ciclo tem que ser executado **pelo menos uma vez**:

**REPETIR**



acções do ciclo

**ENQUANTO (***condição lógica***)**

* A primeira **avaliação da condição** é realizada **depois** de se ter **executado o ciclo** pela primeira vez.
* Se a condição lógica for **verdadeira**, o ciclo **é repetido**.



**APROG - LETI**

Caso prático

* “Adivinhe o número.”

numero\_oculto = 5



**REPETIR**

LER(tentativa)

**ENQUANTO(**tentativa <> numero\_oculto**)**

ESCREVER("Acertou!")



**APROG - LETI**

ENQUANTO … FIM ENQUANTO

* Esta construção é utilizada nos casos em que o ciclo tem que ser executado **sempre sob a condição**:

**ENQUANTO (***condição lógica***)**



acções do ciclo

**FIM ENQUANTO**

* A **primeira execução** do ciclo **depende** da avaliação prévia da **condição**: o ciclo pode nunca ser executado!
* Se a condição lógica for **verdadeira**, o ciclo **é repetido**.



**APROG - LETI**

Caso prático

* “Encher garrafões de 5 litros com limonada”



limonada = 3

garrafoes = 0

**ENQUANTO(**limonada >= 5**)** garrafoes = garrafoes + 1 limonada = limonada - 5

**FIM ENQUANTO**

ESCREVER(garrafoes)



**APROG - LETI**

Ciclos controlados por um contador



**APROG - LETI**

Ciclos controlados por um contador

•

•

Há situações em se sabe o número de vezes que um ciclo vai executar, antes de se iniciar um ciclo.

Nestes casos, há um contador que vai percorrer um conjunto de valores. A dimensão desse conjunto define o número de iterações do ciclo.



**APROG - LETI**

PARA … ATÉ

* Neste construção, é necessário uma variável que vai tomar uma sucessão de valores, desde o inicial até ao final. Essa sucessão define o número de iterações do ciclo.

**PARA** contador **=** *val\_inicial* **ATÉ** *val\_final* **PASSO** *passo* acções do ciclo



**FIM PARA**

* A evolução do contador progride em **incrementos** de **uma unidade**, pelo que é facultativo indicar o passo.
  + Tem que se indicar o passo se for diferente de 1.



**APROG - LETI**

PARA … ATÉ

* Exemplo: apresentar os números pares até 10.

**PARA** i = *2* **ATÉ** *10* **PASSO** *2*



ESCREVER(i)

**FIM PARA**



**APROG - LETI**

Composição de ciclos



**APROG - LETI**

Composição de ciclos

* Podemos compor ciclos, quando **dentro** de um ciclo decorre um bloco de acções que se repetem, **dependendo de outra condição ou contador**.

•

•

Dentro de um ciclo, insere-se um novo ciclo.

O ciclo interior é **inteiramente executado dentro** de cada iteração do ciclo exterior.



**APROG - LETI**

Exemplo

* Simule um relógio, com indicação das horas e minutos.



**APROG - LETI**

Exemplo (ciclos condicionais)

* Simule um relógio, com indicação das horas e minutos.



horas = 0

**ENQUANTO(**horas < 24**)**

minutos = 0

**ENQUANTO(**minutos < 60**)**

**ESCREVE(**horas, minutos**)**

minutos = minutos + 1

**FIM ENQUANTO**

horas = horas + 1

**FIM ENQUANTO**

**APROG - LETI**

Exemplo (ciclos contados)

* Simule um relógio, com indicação das horas e minutos.

**PARA** horas = *0* **ATÉ** *23*



**PARA** minutos = *0* **ATÉ** *59*

**ESCREVE(**horas, minutos**)**

**FIM PARA**

**FIM PARA**



**APROG - LETI**

Implementação em C



**APROG - LETI**

REPETIR … ENQUANTO

•

•

Em C, utiliza-se a construção *do-while*.

**do**



{

*// ciclo*

}

**while(***condição***)**;

O ciclo é sempre realizado pela primeira vez.

* A repetição do ciclo ocorre sempre que o valor da condição seja

VERDADEIRO.



**APROG - LETI**

REPETIR … ENQUANTO

* Exemplo: adivinhe o número.

int numero\_oculto, tentativa;



numero\_oculto = 5;

**do {**

printf("Insira um número até 10: "); scanf("%d", &tentativa);

**} while(**tentativa != numero\_oculto**)**; printf("Acertou!\n");



**APROG - LETI**

ENQUANTO … FIM ENQUANTO

* Em C, utiliza-se a construção *while*.

**while(***condição***)**



{

*// ciclo*

}

* A execução do ciclo depende **sempre** do valor da condição:
* O ciclo só é executado/repetido quando o valor da condição é

VERDADEIRO.



**APROG - LETI**

ENQUANTO … FIM ENQUANTO

* Exemplo: encher garrafões de 5 litros com limonada.

int limonada, garrafoes;



limonada = 3;

garrafoes = 0;

**while(**limonada >= 5**) {**

garrafoes = garrafoes + 1;

limonada = limonada - 5;

**}**

printf("%d garrafões cheios.\n", garrafoes);



**APROG - LETI**

PARA … ATÉ

* Em C, utiliza-se a construção *for*.

**for(***inicialização; condição\_pré\_iteração; operação\_final\_iteração***)** {



*// ciclo*

}

•

•

A execução do ciclo depende **sempre** do valor da condição:

O ciclo só é executado/repetido quando o valor da condição é VERDADEIRO.



**APROG - LETI**

FAZER … ATÉ

* Exemplo: apresentar os números pares até 10.

**int i;**



**for(**i = 2; i <= 10; i = i + 2**)**

{

printf("Número par: %d\n", i);

}



**APROG - LETI**