Comandos de repetição

Disciplina de Programação de Computadores I
Universidade Federal de Ouro Preto

Agenda

- Comando while
- Comando do...while
- Comando for
- Comandos continue e break

Comandos de Repetição

- Comandos de repetição permitem que se execute um bloco de comandos mais de uma vez
    - Ex: Como imprimir os números de 1 a 10?
      Temos que repetir o comando print 10 vezes.

- Os comandos de repetição, em C, são:
    - while
    - do ... while
    - for

Comando while

- O comando while executa um bloco de comando enquanto (e apenas se) uma condição for verdadeira (diferente de 0).

Sintaxe:

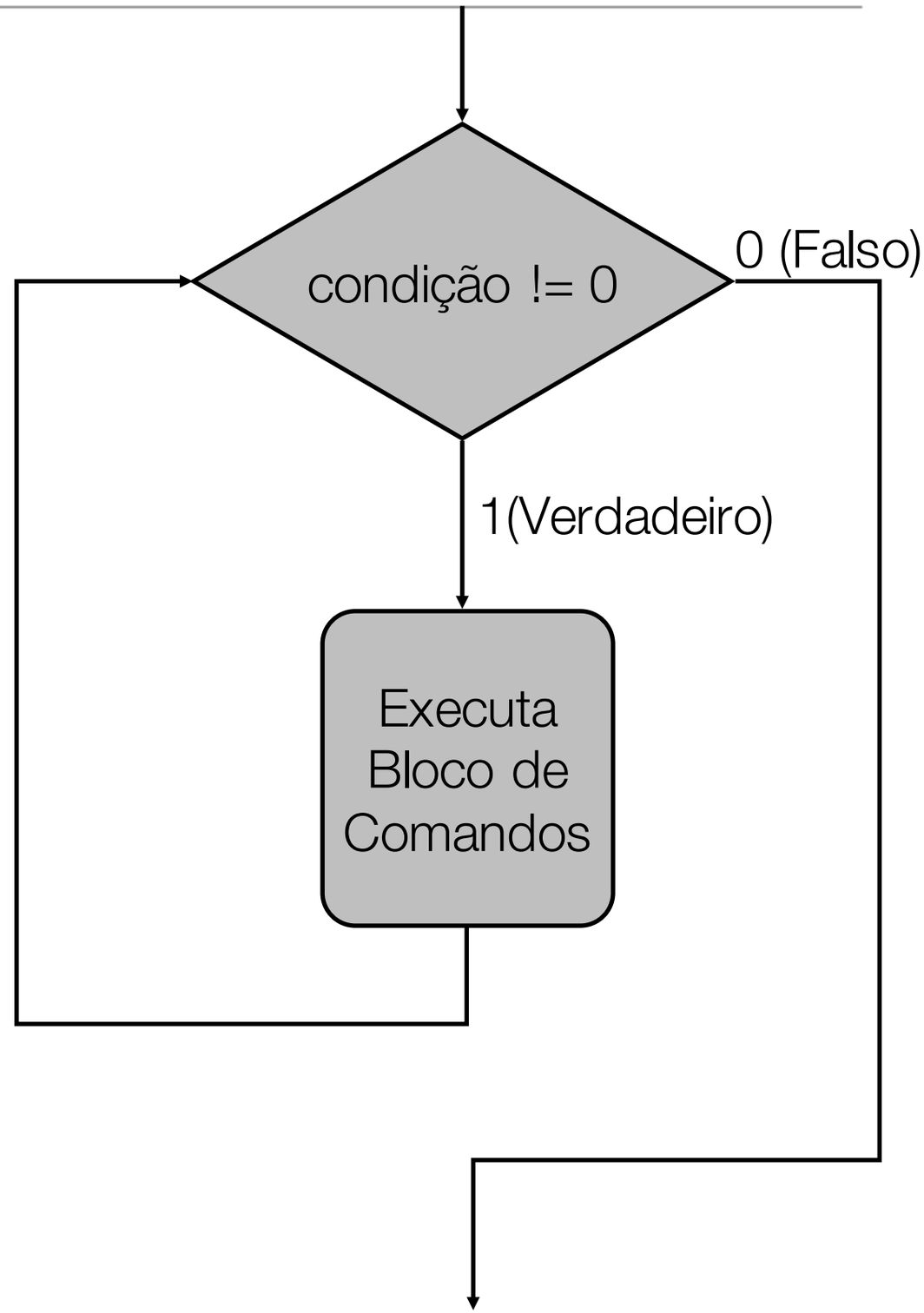
```
while ( condição )
    comando;
```

Ou:

```
while ( condição ){
    comando1;
comando2;...
}
```

Execução do comando while

1. Testa-se a condição é verdadeira
    1. Se a condição é verdadeira:
        1. Executa o bloco de comandos
        2. Volta para 1.
2. Se a condição é falsa:
    1. Sai do while.



Condição extrema I

- Quando a condição é sempre falsa (condição == 0):

```
while (0)
    printf(“Nunca entro!”);
```

- Quando a condição é sempre verdadeira (condição != 0):

```
while (1)
    printf(“Entro no laço e nunca saio! Loop infinito!”);
```

Exemplo: Imprimir os números de 1 a 10

```
int i = 1;

while (i <=10){

    printf (“%d\n”, i);

    i++;

}
```

Exemplo: Imprimir os números de 1 até um dado n

```
int i = 1;

int n = 0;

scanf(“%d”,&n);

while(i <=n){

      printf (“%d\n”, i);

      i++;

}
```

Comando do...while

- O comando do...while executa um bloco de comandos e continua sua execução enquanto uma condição for verdadeira.

Sintaxe:

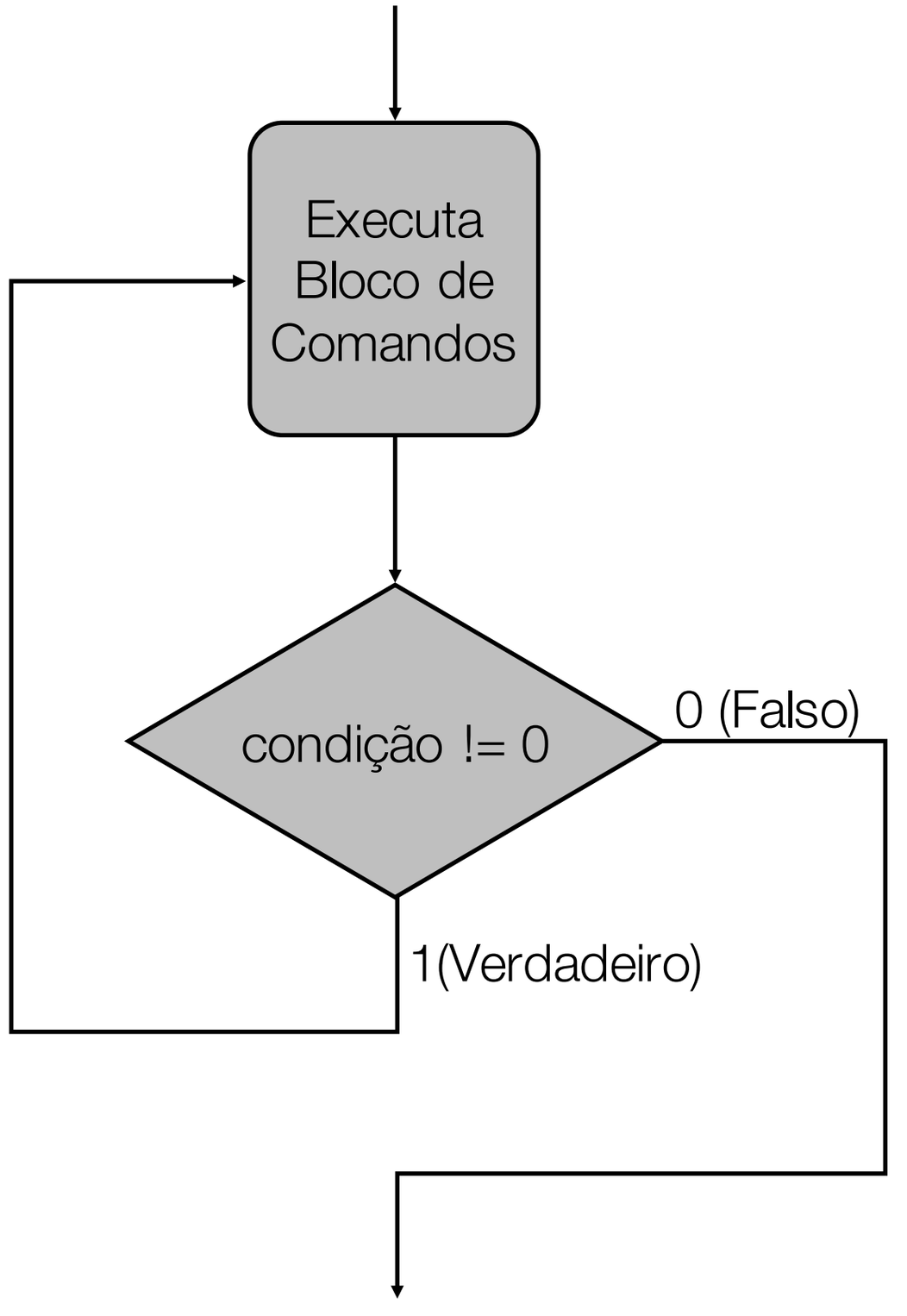
```
do
    comando;
while (condição);
```

Ou:

```
do{
    comando1;
    comando2; ...
} while (condição);
```

Execução do comando while

1. Executa o bloco de comandos.
2. Testa-se a condição é verdadeira
    1. Se a condição é verdadeira:
        1. Volta para 1.
    2. Se a condição é falsa:
        1. Sai do while.



Exemplo: Imprimir os números de 1 a 10

```
int i = 1;
do{
    print (“%d\n”, i);
    i++;
} while(i <=10);
```

Exemplo: Imprimir os números de 1 até um dado n

```
int i = 1;

int n = 0;

scanf(“%d”,&n);

do{

      print (“%d\n”, i);

      i++;

} while(i <=n);
```

Condição extrema II

- Quando a condição for falsa antes de chegar ao teste da condição:

do...while sempre executará uma vez o bloco de comando.

while apenas executará o bloco se a condição for verdadeira.

do...while

```
int i = 1; int n = 0;

do{
    printf (“%d\n”, i);
    i++;
} while(i <=n);
```

while

```
int i = 1; int n = 0;

while(i <=n){
    printf (“%d\n”, i);
    i++;
}
```

Comando for

- O comando for executa um bloco de comandos um número determinado de vezes

Sintaxe:

```
for ( início; teste; incremento)
    comando;
```

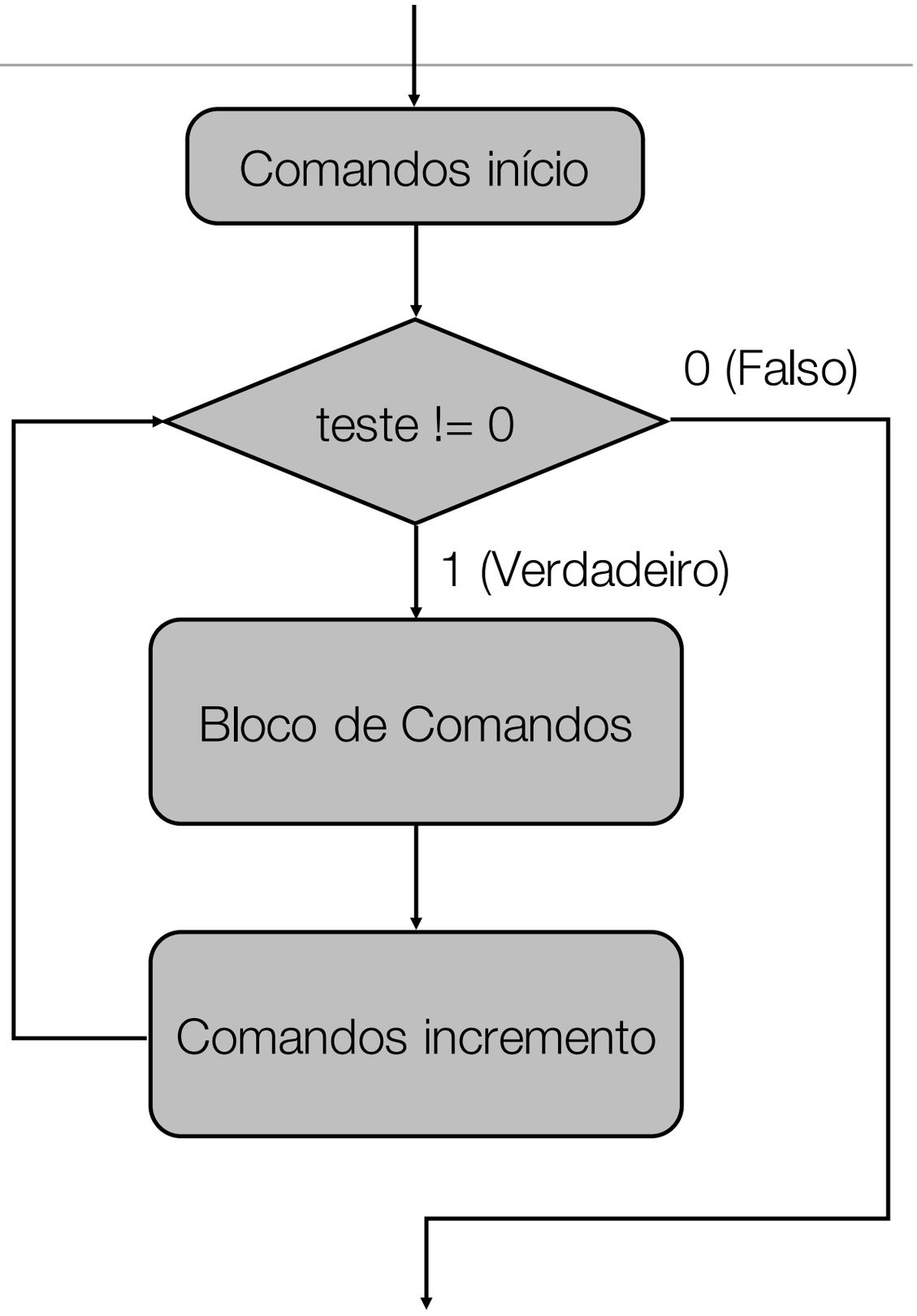
Ou:

```
for ( início; teste; incremento){
    comando1;
    comando2;...
}
```

início: atribuições iniciais de variáveis, separadas por ‘,’
teste: teste a ser realizado para continuar o laço
incremento: um ou mais incrementos de variáveis, separados por ‘,’

Execução do comando for

1. Executa as atribuições do início.
2. Executa o teste.
    1. Se for verdadeiro:
        1. Vai para 3.
    2. Se for falso:
        1. Sai do for.
3. Executa o bloco de comandos (ou corpo) do for.
4. Executa os comandos em incremento.
5. Volta para 2.



Equivalência entre for e while

```
for ( início; teste; incremento ){
    comando1;
    comando2;...
}
```

```
início;
while ( teste ){
    comando1;
    comando2; ...
    incremento;
}
```

Exemplo: Imprimir os números de 1 a 10

```
int i = 0;

for( i = 1; i <= 10; i = i+1){

    printf ("\n %d",i);

}
```

Exemplo: Imprimir os números de 1 até um dado n

```
int i = 0;

int n = 0;

scanf(“%d”,&n);

for ( i = 1; i <= n; i++ ){

    printf (“%d\n”, i);

}
```

Comando break e os laços

- O comando **break** termina a execução de um laço, passando a execução para o próximo comando após o laço.

```
int i;
for(i = 1; i<= 10 ; i++){
      if(i >= 5)
            break;
      printf("%d\n",i);
}
printf("Fim do laço.\n");
```

Saída:

```
1
2
3
4
Fim do laço.
```

Utilizando break no lugar do teste do for

- Pode-se escrever um for em que o teste está no bloco de comandos utilizando o **break**.

- Os códigos abaixo são equivalentes:

```
int i;
for(i = 1;      ; i++){
     if(i > 10)
          break;
     printf("%d\n",i);
}
```

```
int i;
for(i = 1; i<=10 ; i++){
     printf("%d\n",i);
}
```

O comando continue

- O comando **continue** faz a execução pular os próximos comandos e ir direto para o fim do laço (possivelmente repetindo o laço se o teste para continuação for verdadeiro).

```
int i;
for(i = 1; i<= 10 ; i++){
      if(i == 5)
            continue;
      printf("%d\n",i);
}
printf("Fim do laço.\n");
```

Saída:

1
2
3
4
6
7
8
9
10
Fim do laço.

Uso do comando continue

- O comando **continue** é utilizando quando se deseja que comandos dentro do laço sejam executados apenas se uma condição for verdadeira.

```
int i;
for (i = 0; i <= 10; i++){
    if( (i % 2) != 0)
        continue;
    printf("%d\n", i);
}
```

```
int i;
for (i = 0; i <= 10; i+=2){
    printf("%d\n", i);
}
```

Saída:

```
0
2
4
6
8
10
```

Referências Bibliográficas

- Material de aula do Prof. Ricardo Anido, da UNICAMP: http://www.ic.unicamp.br/~ranido/mc102/

- Material de aula da Profa. Virgínia F. Mota: https://sites.google.com/site/virginiaferm/home/disciplinas

- DEITEL, P; DEITEL, H. *C How to Program*. 6a Ed. Pearson, 2010.