

Estruturas de Repetição

Prof. Gabriel Barbosa da Fonseca

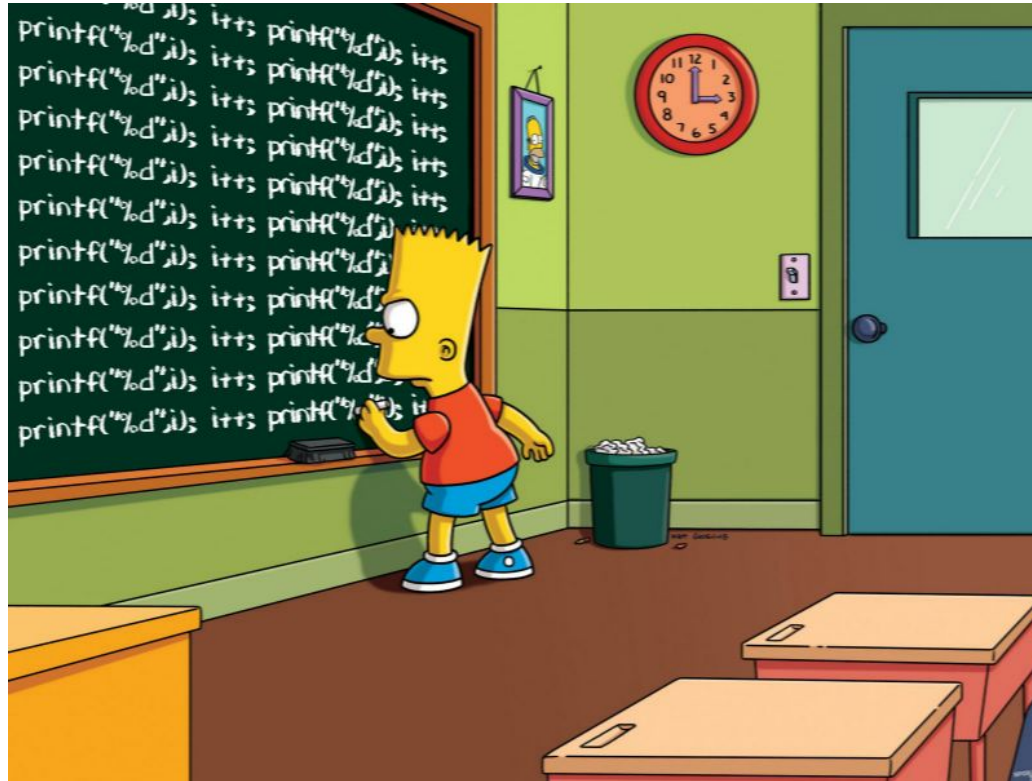
Email: gbfonseca@sga.pucminas.br

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

Estruturas de Repetição

Como imprimir os 1000 primeiros números a partir do 1?

Estruturas de Repetição



Estruturas de Repetição

Como imprimir os 1000 primeiros números a partir do 1?

```
printf("1");  
printf("2");  
printf("3");  
....  
....  
printf("1000");
```

NADA PRÁTICO!

Estruturas de Repetição

Como imprimir os 1000 primeiros números a partir do 1?

- Outra solução seria o uso de uma estrutura de repetição
- Por exemplo, o comando ***Enquanto (while)***

Comando WHILE (enquanto)

enquanto (expressão) **faça**

|

lista de comandos

fim enquanto

Comando WHILE (enquanto)

```
while (expressão) {
```

```
    lista de comandos
```

```
}
```

Comando WHILE (enquanto)

```
while (expressão) {
```

```
    lista de comandos
```

```
}
```


Comando WHILE (enquanto)

Como imprimir os 1000 primeiros números a partir do 1?

- Outra solução seria o uso de uma estrutura de repetição
- Por exemplo, o comando ***Enquanto*** (***while***)

```
int i = 1;  
  
while (i <= 1000){  
    printf("%d",i);  
    i++;  
}
```

Comando WHILE (enquanto)

```
int i = 1;  
  
while (i <= 1000){  
    printf("%d",i);  
    i++;  
}
```

Comando WHILE (enquanto)

```
int i = 1;
```



Verdadeiro?

```
while (i <= 1000){  
    printf("%d",i);  
    i++;  
}
```

Comando WHILE (enquanto)

```
int i = 1;
```

```
while (i <= 1000){  
    printf("%d",i);  
    i++;  
}
```



SIM



Executa

Comando WHILE (enquanto)

```
int i = 1;
```



Verdadeiro?

```
while (i <= 1000){  
    printf("%d",i);  
    i++;  
}
```

Comando WHILE (enquanto)

```
int i = 1;
```

NÃO

```
while (i <= 1000){  
    printf("%d",i);  
    i++;  
}
```

Pula

Execução do comando While

```
int i = 1;
```

```
while (i <= 1000){  
    printf("%d",i);  
    i++;  
}
```

Memória	
i	
1	
2	
3	
4	
5	
...	
...	
999	
1000	
1001	

Tela
1
2
3
4
5
...
...
999
1000

Exercício

Faça o quadro de memória e saída na tela para o código abaixo

```
int i= 2;  
  
while(i<=10){  
    printf("\n%d",i-1);  
    i+=3;  
}
```

Memória		Tela
i		

Exercício

Faça o quadro de memória e saída na tela para o código abaixo

```
int i= 2;  
  
while(i<=10){  
    printf("\n%d", (i-1));  
    i+=3;  
}
```

Memória	
i	
2	
5	
8	
11	

Tela
1
4
7

Estrutura Faça-Enquanto (DO-WHILE)

- Similar ao comando **enquanto**
- A diferença entre eles é o **momento em que a expressão é avaliada**
- No **while**, a expressão é avaliada antes da execução dos comandos
- No faça-enquanto (**do-while**), a lista de comandos é executada e, **depois**, a expressão é avaliada

Estrutura Faça-Enquanto (DO-WHILE)

faça



lista de comandos

enquanto (expressão);

Estrutura Faça-Enquanto (DO-WHILE)

do {

lista de comandos

} while (expressão);

Estrutura Faça-Enquanto (DO-WHILE)

```
do {
```

```
    lista de comandos
```

```
} while (expressão);
```

Estrutura Faça-Enquanto (DO-WHILE)

```
int i= 100;  
while(i<=10){  
    printf("\n%d",i);  
    i+=10;  
}
```

```
int i= 100;  
do{  
    printf("\n%d",i);  
    i+=10;  
}while(i<=10);
```

Exercício

Leia dois números reais e realize a divisão entre eles:

Exercício

Leia dois números reais e realize a divisão entre eles:

```
double num1, num2;

escrever: Digite o primeiro número: ";
ler num1;

do {
    escrever: Digite o segundo número: ";
    ler num2;
}while (num == 0);

escrever: Divisao: ", (num1 / num2);
```


Exercício

Leia um número inteiro e garanta que ele é um mês válido

Exercício

Leia um número inteiro e garanta que ele é um mês válido

```
int mes;  
  
do {  
    escrever: "Insira um mês válido: "  
    ler mes;  
} while ((mes >= 1 && mes <= 12) == false);
```

Estrutura – Para (FOR)

- Similar ao comando enquanto, contudo, ele permite: pré-comandos (início) e pós-comandos (incremento)

Estrutura – Para (FOR)

- Similar ao comando enquanto, contudo, ele permite: pré-comandos (início) e pós-comandos (incremento)

```
início;
```

```
while (expressão) {  
    lista de comandos;
```

```
    incremento;
```

```
}
```

```
int i = 1;
```

```
while (i <= 1000){  
    escrever: i;
```

```
    i++;
```

```
}
```

Estrutura – Para (FOR)

para (início ; expressão ; incremento)

|

lista de comandos

fim para

Estrutura – Para (FOR)



para (início ; expressão ; incremento)

lista de comandos

fim para

Estrutura – Para (FOR)

para (**início** ; expressão ; incremento)

lista de comandos

fim para

Estrutura – Para (FOR)



```
para (início ; expressão ; incremento)
```

lista de comandos

fim para

Estrutura – Para (FOR)

para (início ; expressão ; incremento)

lista de comandos

fim para

Verdadeiro?

Estrutura – Para (FOR)

para (início ; expressão ; incremento)

TRUE

lista de comandos

fim para

Estrutura – Para (FOR)

para (início ; expressão ; incremento)



lista de comandos

fim para

Estrutura – Para (FOR)

para (início ; expressão ; incremento)

lista de comandos

fim para



Estrutura – Para (FOR)



```
para (início ; expressão ; incremento)
```

lista de comandos

fim para

Estrutura – Para (FOR)

para (início ; expressão ; incremento)

lista de comandos

fim para

Verdadeiro?

Estrutura – Para (FOR)

para (início ; expressão ; incremento)

TRUE

lista de comandos

fim para

Estrutura – Para (FOR)

para (início ; expressão ; incremento)



lista de comandos

fim para

Estrutura – Para (FOR)

para (início ; expressão ; **incremento**)

lista de comandos

fim para



Estrutura – Para (FOR)



```
para (início ; expressão ; incremento)
```

lista de comandos

fim para

Estrutura – Para (FOR)

para (início ; expressão ; incremento)

lista de comandos

fim para

Verdadeiro?

Estrutura – Para (FOR)

para (início ; expressão ; incremento)

FALSE

lista de comandos

fim para

Estrutura – Para (FOR)

para (início ; expressão ; incremento)

lista de comandos

fim para

Termina a repetição

Sintaxe em C/C++ – FOR

```
for (início ; expressão ; incremento) {  
    lista de comandos  
}
```

Sintaxe em C/C++ – FOR

- Como fazer um programa usando o comando FOR para escrever os números inteiros de 1 a 3?

```
for ( int i = 1 ; i <= 3 ; i++){  
    printf("%d ",i);  
}
```

Sintaxe em C/C++ – FOR

```
for ( int i = 1 ;   i <= 3 ;   i++){  
    printf("%d ",i);  
}
```

TELA

QUADRO DE MEMÓRIA

i	
---	--

Sintaxe em C/C++ – FOR

```
for ( int i = 1 ; i <= 3 ; i++){  
    printf("%d ",i);  
}
```

TELA

1
2
3

QUADRO DE MEMÓRIA

i	1	2	3	4

Exercício

- Faça um programa para ler um número inteiro **N** e mostrar na tela os **N** primeiros números inteiros e positivos

Exercício

- Faça um programa para ler um número inteiro **N** e mostrar na tela os **N** primeiros números inteiros e positivos

```
int n;  
  
ler n;  
  
for (int i = 1; i <= n; i += 1){  
    escrever: i;  
}
```

Exercício

- Faça um programa para imprimir os quatro primeiros múltiplos de cinco

Exercício

- Faça um programa para imprimir os quatro primeiros múltiplos de cinco

```
for ( int i = 0, valor = 5 ; i < 4 ; i += 1, valor += 5 ){  
    printf("%d\n",valor);  
}
```

Exercício

O que será mostrado na tela???

```
for ( int i = 0, valor = 5 ; i < 4 ; i += 1, valor += 5 ){  
    printf("%d\n",valor);  
}  
printf("Valor = %d",valor);
```

Exercício

```
for ( int i = 0, valor = 5; i < 4 ;i += 1, valor += 5 ){  
    printf("%d\n",valor);  
}  
printf("Valor = %d",valor);
```

main.c:7:21: error: use of undeclared identifier 'valor'

printf("Valor = %d",valor);

1 error generated.

make: *** [<builtin>: main.o] Error 1

exit status 2

Exercício

E agora???

```
int valor = 5;
for ( int i = 0; i < 4 ; i += 1, valor += 5 ){
    printf("%d\n",valor);
}
printf("Valor = %d",valor);
```


Exercício

Saída:

5

10

15

20

Valor = 25

```
int valor = 5;
for ( int i = 0; i < 4 ; i += 1, valor += 5 ){
    printf("%d\n",valor);
}
printf("Valor = %d",valor);
```

Exercício

- Faça um programa para ler dois números inteiros e multiplicá-los sem utilizar a operação de multiplicação

Exercício

- Faça um programa para ler dois números inteiros e multiplicá-los sem utilizar a operação de multiplicação

Algoritmo

```
int n1, n2, i, soma;  
escrever: "Digite o multiplicando: ";  
ler n1;  
escrever: "Digite o multiplicador: ";  
ler n2;  
for(int i = 1, soma = 0; i <= n2; i++, soma = soma + n1);  
escrever: "Produto: ", soma;
```

Fim Algoritmo