

Aula 5: Estruturas de Repetição

Professor(a): João Eduardo Montandon (103)

Virgínia Fernandes Mota (106)

jemaf.github.io

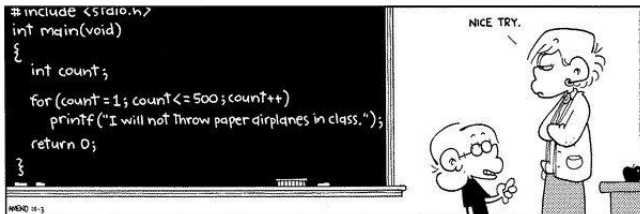
<http://www.dcc.ufmg.br/~virginiaferm>

INTRODUÇÃO A PROGRAMAÇÃO - SETOR DE INFORMÁTICA



Introdução

- Em vários momentos torna-se necessário repetir um trecho (ou bloco) de comandos por um certo número de vezes.
 - Esses laços são conhecidos como "loops".
- Ao invés de se replicar trechos de código, usamos "loops" para repetí-los:
 - Economia de espaço, facilidade de programação e quantidade de repetições pode ser definida em tempo de execução!



Tipos de repetição

- Repetição com teste no início.
- Repetição com teste no fim.
- Repetição com variável de controle.

Repetição com teste no início

```
1 ...  
2 while (condição){  
3     bloco de operações  
4 }  
5 ...
```

- Semântica: Enquanto a condição for verdadeira, a sequência será repetida. Quando a condição fornecer resultado falso, o controle sai da estrutura passando para o comando seguinte ao final do bloco.

- Exemplo 1: Uso de FLAG
 - FLAG é um valor específico fornecido após o último dado de entrada, que serve para indicar o fim dos dados de entrada.
 - FLAG é somente uma marca de fim dos dados de entrada (não é um dado de entrada) e não pode ser processado.
 - A leitura do FLAG informa ao programa que os dados de entrada terminaram e que ele deve partir para a execução da finalização de seu processamento (cálculos finais, impressões finais etc.).

Desenvolver um algoritmo para ler uma sequência de números inteiros, calcular e imprimir o quadrado de cada número lido. O último valor a ser lido é um $\text{FLAG} = 0$.

Repetição com teste no início

```
1 #include <stdio.h>
2 int main() {
3     int num, quadrado;
4     printf("Digite um numero inteiro: ");
5     scanf("%d", &num);
6     while( num != 0) {
7         quadrado = num * num;
8         printf("\n O quadrado de %d é %d:", num, quadrado);
9         printf("\n Digite um numero inteiro: ");
10        scanf("%d", &num);
11    }
12    return 0;
13 }
```

- Exemplo 2: Uso de acumulador
 - Uma forma possível para resolver o problema de somatório de valores (acumulador) é imaginar uma variável que armazena as somas parciais. Essa variável deve iniciar com zero e somar número por número, até o final da sequência.

Desenvolver um algoritmo para ler uma sequência de números inteiros com $FLAG = 0$, calcular e imprimir a soma desses números

Repetição com teste no início

```
1 #include <stdio.h>
2 int main() {
3     int num, soma;
4     printf("Digite um numero inteiro: ");
5     scanf("%d", &num);
6     soma = 0;
7     while( num != 0) {
8         soma = soma + num;
9         printf("\n Digite um numero inteiro: ");
10        scanf("%d", &num);
11    }
12    printf("A soma dos numeros digitados foi %d ", soma);
13    return 0;
14 }
```


- Exemplo 3: Uso de contador
 - Contador é um acumulador que é incrementado de 1 em 1.

Desenvolver um algoritmo para ler uma sequência de números inteiros com $\text{FLAG} = 0$, calcular e imprimir a quantidade de números lidos.

Repetição com teste no início

```
1 #include <stdio.h>
2 int main() {
3     int num, cont;
4     printf("Digite um numero inteiro: ");
5     scanf("%d", &num);
6     cont = 0;
7     while( num != 0) {
8         cont++; // cont = cont + 1
9         printf("\n Digite um numero inteiro: ");
10        scanf("%d", &num);
11    }
12    printf("Foram lidos %d numeros", cont);
13    return 0;
14 }
```

Repetição com teste no final

```
1 ...  
2 do{  
3     bloco de operações  
4 }while (condição);  
5 ...
```

- Semântica: Enquanto a condição for verdadeira, a sequência será repetida. Quando a condição fornecer resultado falso, o controle sai da estrutura passando para o comando seguinte ao final do bloco.

Exemplo 1:

Desenvolver um algoritmo para ler 100 números inteiros, calcular e imprimir o quadrado de cada número lido.

Repetição com teste no final

```
1 #include <stdio.h>
2 int main() {
3     int num, cont;
4     cont = 0;
5     do{
6         printf("\n Digite um numero inteiro: ");
7         scanf("%d", &num);
8         printf("\n O quadrado de %d eh %d ", num, num * num);
9         cont++; // cont = cont + 1
10    }while(cont < 100);
11    return 0;
12 }
```

Repetição com teste no final

Exemplo 2: Qual a diferença?

```
1 ...  
2     int A;  
3     A = 0;  
4     while (A < 10){  
5         printf("%d", A);  
6         A++;  
7     }  
8     ...
```

```
1 ...  
2     int A;  
3     A = 0;  
4     do{  
5         printf("%d", A);  
6         A++;  
7     }while (A < 10);  
8     ...
```

Repetição com variável de controle

```
1 ...  
2 for (V = I; V <= L; V = V + P){  
3     bloco de operações  
4 }  
5 ...
```

- V é a variável de controle
- I é o valor inicial de V
- L é o valor limite de V
- P é o incremento sofrido por V após cada execução do bloco de operações.

Repetição com variável de controle

```
1 ...  
2 for (V = I; V <= L; V = V + P){  
3     bloco de operações  
4 }  
5 ...
```

- Semântica: o controle de fluxo de execução entra na estrutura e faz a etapa de inicialização ($V = I$) iniciando a estrutura de repetição na seguinte sequência:
 - 1 executa o teste ($V \leq L$). Se for válido, vai para o passo seguinte. Senão, executa o primeiro comando após a estrutura de repetição;
 - 2 executa o bloco de operações;
 - 3 executa o incremento ($V = V + P$);
 - 4 retorna ao passo 1.

Repetição com variável de controle

Exemplo 1:

Imprimir os valores inteiros que se encontram entre um valor inicial e um valor final.

Repetição com variável de controle

```
1 #include <stdio.h>
2 int main(){
3     int num_inicial, num_final, i;
4     printf("Digite o numero inicial: ");
5     scanf("%d", &num_inicial);
6     printf("Digite o numero final: ");
7     scanf("%d", &num_final);
8     for( i = num_inicial; i <= num_final; i++) {
9         printf("valor de i:%d \n", i);
10    }
11    return 0;
12 }
```

Repetição com variável de controle

Exemplo 2:

Fazer um algoritmo para calcular e imprimir a tabuada de multiplicação do número TAB entre 1 e NUM.

Repetição com variável de controle

```
1 #include <stdio.h>
2 int main(){
3     int i, num, tab;
4     printf("Tabuada de: ");
5     scanf("%d", &tab);
6     printf("Ate que numero: ");
7     scanf("%d", &num);
8     for( i = 1; i <= num; i++) {
9         printf("%d x %d = %d \n", i, tab, i*tab);
10    }
11    return 0;
12 }
```

Tipos de repetição

- Repetição com teste no início : **while**
- Repetição com teste no fim : **do-while**
- Repetição com variável de controle: **for**

Como sair de um loop prematuramente??

- **break** : Força o loop a sair imediatamente.
- **continue**: Pula para a próxima iteração.
- **exit**: Finaliza o programa.

1. Elabore um programa que calcule $N!$ (fatorial de N), sendo que o valor inteiro de N é fornecido pelo usuário.
2. Escrever um algoritmo que lê um valor N inteiro e positivo e que calcula e escreve o valor de E : $E = 1 + 1/2! + 1/3! + \dots + 1/N!$
3. Faça um programa que, dado um conjunto de valores inteiros e positivos (fornecidos um a um pelo usuário), determine qual o menor e o maior valor do conjunto. O final do conjunto de valores é conhecido através do valor zero, que não deve ser considerado.
4. Fazer um programa para calcular e mostrar os N primeiros termos da série de Fibonacci. O número N é fornecido pelo usuário. A série de Fibonacci é : 1 1 2 3 5 8 13 ... Isto é $f_1 = f_2 = 1$, $f_3 = f_1 + f_2$, $f_4 = f_2 + f_3$...
5. A conversão de graus Fahrenheit para Centígrados é obtida pela fórmula $C = 9*(F-32)/5$. Escreva um programa que calcule e escreva uma tabela de graus centígrados em função de graus Fahrenheit que variem de 50 a 150 de 1 em 1.

Na próxima aula...

Prova!