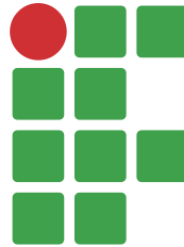


Operadores Lógicos e Estrutura de Repetição WHILE



**INSTITUTO
FEDERAL**
Paraíba

Professor: Paulo de Tarso F. Júnior
paulodt@gmail.com

Operadores Lógicos

- ▶ A linguagem C fornece operadores lógicos que podem ser usados na criação de condições mais . complexas, combinando condições simples:
 - ▶ (E Lógico) → `&&`
 - ▶ (OU lógico) → `||`
 - ▶ (NÃO lógico também chamado negação) → `!`

Operadores Lógicos - &&

- ▶ Assegurar que duas condições sejam verdadeiras AO MESMO TEMPO antes de escolher determinado caminho de execução.

```
if (sexo == 1 && idade >= 65)
```

```
    idosas = idosas + 1;
```

Operadores Lógicos - ||

- ▶ Assegurar de que, em algum ponto do programa, PELO MENOS UMA de duas condições seja verdadeira

```
if (mediaSemestre >= 90 || final >= 90)  
    printf("O conceito do aluno e A\n");
```

Operadores Lógicos - !

- ▶ Operador Unário
- ▶ O operador de negação lógica é colocado antes de uma condição quando estamos em invertê-la:

```
if (!(conceito == valorSentinela))  
    printf("Escreva a saída %f\n", saida);
```

Operadores de Atribuição

- ▶ Operadores de Atribuição → Abreviação de expressões de atribuição
 - ▶ Exemplo em C
 - ▶ `c = c + 3;`
 - ▶ Abreviação a partir do operador de atribuição de adição
 - ▶ `c += 3;`
 - ▶ Forma de atribuição
 - ▶ *variável = expressão do operador de variável;*
 - ▶ Re-escrita
 - ▶ *operador de variável = expressão;*

Operadores de Atribuição

- ▶ Exemplos de outros operadores de atribuição
 - ▶ $d -= 4$ ($d = d - 4$)
 - ▶ $e *= 5$ ($e = e * 5$)
 - ▶ $f /= 3$ ($f = f / 3$)
 - ▶ $g \% = 9$ ($g = g \% 9$)

Operadores de Incremento/Decremento

- ▶ Operador de *Incremento* (++)
 - ▶ Uso em substituição de +=1
- ▶ Operador de *Decremento* (--)
 - ▶ Uso em substituição de -=1
- ▶ *Pré-incremento/ Pré-decremento*
 - ▶ Operador usado *antes da variável* (++c ou --c)
 - ▶ Variável alterada (incrementada/ decrementada) *antes da* avaliação da expressão

Operadores de Incremento/Decremento

- ▶ *Pós-incremento/Pós-decremento*
 - ▶ Operador usado após da variável (*c++ ou c--*)
 - ▶ Expressão executada *antes da alteração* (incremento/decremento) da variável
- ▶ Se c igual a 5
 - ▶ `printf("%d", ++c);`
 - ▶ Imprimirá 6
 - ▶ `printf("%d", c++);`
 - ▶ Imprimirá 5
 - ▶ Em todo caso, c assume agora o valor 6

Operadores de Incremento/Decremento

- ▶ Quando a variável *não está numa expressão*
 - ▶ Pré-incremento e pós-incremento apresentam mesmo efeito
 - ▶ `++c;`
 - ▶ `printf("%d", c);`
 - ▶ Apresenta o mesmo efeito de
 - ▶ `c++;`
 - ▶ `printf("%d", c);`

Estrutura de Repetição WHILE

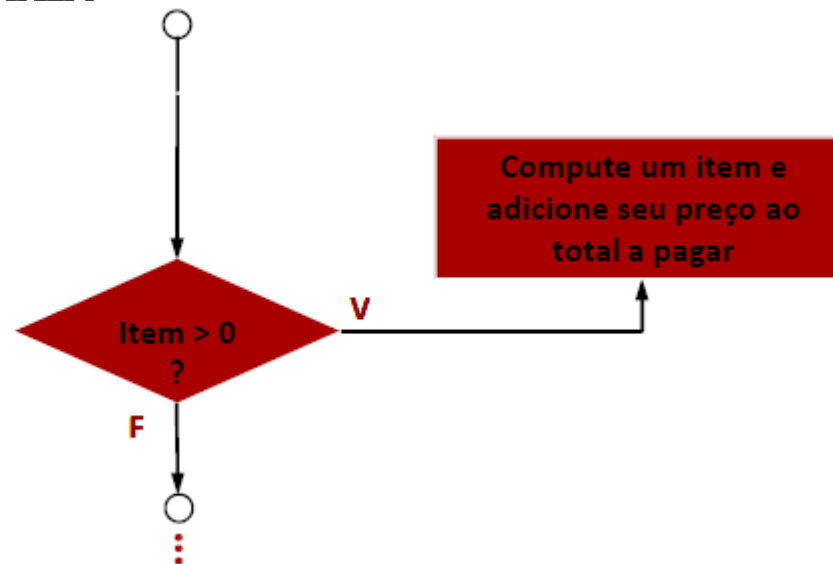
- ▶ Estrutura de Repetição
 - ▶ Especificação de uma ação a ser repetida enquanto uma dada condição permanecer *Verdadeira*
 - ▶ Exemplo de pseudocódigo
 - ▶ *Enquanto houver itens no carrinho de compras*
 - ▶ *Retire um item e adicione seu preço ao somatório de itens já computados*
 - ▶ O laço *while* será repetido até que a condição imposta se torne *falsa*

Estrutura de Repetição WHILE

► Exemplo

while(item > 0)

*total_a_pagar = total_a_pagar +
custo_item*



Estrutura de Repetição WHILE

- ▶ Repetição controlada por contador
 - ▶ Laço repetido até que o contador atinja um determinado limiar (pré-definido pelo programador)
 - ▶ Repetição Delimitada → Número de repetições é conhecido
 - ▶ Exemplo
 - ▶ Uma turma com 40 alunos fez uma prova. As notas da prova (números reais com uma casa decimal, na faixa de 0,0 a 10,0) estão disponíveis. Determinar a média da turma na referida prova.

Estrutura de Repetição WHILE

```
1 /* Programa 01 - Média de uma turma de 40 alunos usando  
   repetição controlada por contador */
```

```
4 #include <stdio.h>
```

```
5
```

```
6 int main()
```

```
7 {
```

```
8 int contador;
```

```
9 float nota, total, media;
```

```
10 /* fase de inicialização */
```

```
11 total = 0;
```

```
12 contador = 1;
```

```
13
```

```
14 /* fase de processamento */
```

```
15 while (contador <= 40) {
```

```
16     printf("Digite nota: ");
```

```
17     scanf("%f", &nota);
```

```
18     total = total + nota;
```

```
19     contador = contador + 1;
```

```
20 }
```

```
21 /* fase de conclusão */
```

```
22 media = total / 40;
```

```
23 printf("A média da turma é %f\n", media);
```

```
24
```

```
25 return 0; /* indica que o programa  
   terminou satisfatoriamente */
```

Estrutura de Repetição WHILE

- ▶ Delineamento do Problema sobre o Exemplo Anterior
 - ▶ Desenvolvimento de um programa para a determinação da média de uma turma, considerando que o programa processará um número arbitrário de notas cada vez que for executado
 - ▶ Número desconhecido de alunos
 - ▶ *Como o programa será finalizado?*

Estrutura de Repetição WHILE

► Solução do Problema

► Uso de um valor “sentinela”

- Também denominado valor de *sinalização*, “*testa de ferro*” (*dummy value*) ou *flag*
- Indicação do final da entrada de dados
- Laço finalizado quando o usuário passa para o programa o valor “*sentinela*”
- Valor sentinela não deve ser confundido com uma entrada regular (e.g. -1 no caso do programa da média das notas)

Estrutura de Repetição WHILE

- ▶ Representação do topo (top) em pseudocódigo
 - ▶ Determine a média da turma
- ▶ Divisão do topo em tarefas menores e listagem ordenada das tarefas
 - ▶ Inicialize as variáveis
 - ▶ Receba as notas digitadas, some-as e conte-as
 - ▶ Calcule e imprima a média da turma

Estrutura de Repetição WHILE

- ▶ Fases de diversos programas
 - ▶ Inicialização
 - ▶ Fase na qual são atribuídos valores iniciais às variáveis do programa
 - ▶ Processamento
 - ▶ Fase de aquisição dos valores dos dados de entrada e de manipulação apropriada das variáveis do programa
 - ▶ Finalização
 - ▶ Fase de cálculo e impressão dos resultados finais

Estrutura de Repetição WHILE

```
1 /* Programa 02 - Média de uma turma
2 com repetição controlada por
3 sentinela */
4 #include <stdio.h>
5
6 int main()
7 {
8 float media, nota, total; /* novo tipo de dados */
9 int contador;
10
11 /* fase de inicialização */
12 total = 0;
13 contador = 0;
14
15 /* fase de processamento */
16 printf( "Digite nota, -1 ao final: " );
17 scanf( "%d", &nota );
18
```

```
19 while ( nota != -1 ) {
20 total = total + nota;
21 contador = contador + 1;
22 printf( "Digite nota, -1 ao final: " )
23 scanf( "%d", &nota );
24 }
25
26 /* fase de finalização */
27 if ( contador != 0 ) {
28 media = ( float ) total / contador; /* casting */
29 printf( "A média da turma é %.2f", media );
30 }
31 else
32 printf( "Nenhuma nota foi digitada\n" );
33
34 return 0; /* indica que o programa
termino
```

Operadores Lógicos e Estrutura de Controle WHILE

Professor: Paulo de Tarso F. Júnior

paulodt@gmail.com

