

Fundação Hermínio Ometto

Bacharelado em Sistemas de Informação

SIF009 - Linguagem de Programação I

Prof. Dr. Sérgio Luis Antonello

Aula 04

25/08/020

Fundação Hermínio Ometto

Bacharelado em Sistemas de Informação

SIF009 - Linguagem de Programação I

Prof. Dr. Sérgio Luis Antonello

Aula 04

25/08/020

Plano de ensino

1. Unidade I - Programação estruturada e Linguagem C (objetivos b, c e d).
 - 1.1. Conceitos de programação estruturada.
 - 1.2. Estrutura de um programa de computador.
 - 1.3. Códigos fonte, objeto e executável.
 - 1.4. Biblioteca de códigos.
 - 1.5. Compiladores e Interpretadores.
 - 1.6. Processos de compilação e link edição.
 - 1.7. Identificação dos tipos de erros e alertas (léxicos, sintáticos e semânticos).
 - 1.8. Depuração de código.
 - 1.9. Palavras reservadas.
 - 1.10. Tipos de dados.
 - 1.11. Constantes. Variáveis simples e estruturadas. Escopo de variáveis.
 - 1.12. Operadores e precedência.
 - 1.13. Expressões aritméticas, lógicas e relacionais.
 - 1.14. Comandos.
 - 1.15. Ambientes de desenvolvimento e programação.
2. Unidade II - Estruturas de controle (sequência, decisão e repetição), registro e arquivo (objetivos a, c, d).
 - 2.1. Comandos `if` e `switch`.
 - 2.2. Comandos `for`, `while` e `do while`.
 - 2.3. Blocos de comandos e aninhamento.
 - 2.4. Definição de tipos.
 - 2.5. Registro.
 - 2.6. Arquivo: leitura e gravação de dados em disco.
3. Unidade III - Ponteiros e Funções (objetivos a, c, d, e).
 - 3.1. Ponteiros.
 - 3.2. Funções.
 - 3.3. Passagem de parâmetro por valor.
 - 3.4. Passagem de parâmetro por referência.
4. Unidade IV - Strings e Variáveis indexadas (objetivos a, c, d).
 - 4.1. Manipulação de strings.
 - 4.2. Manipulação de caracteres.
 - 4.3. Declaração e manipulação de vetores.
 - 4.4. Declaração e manipulação de matrizes.

Sumário da aula

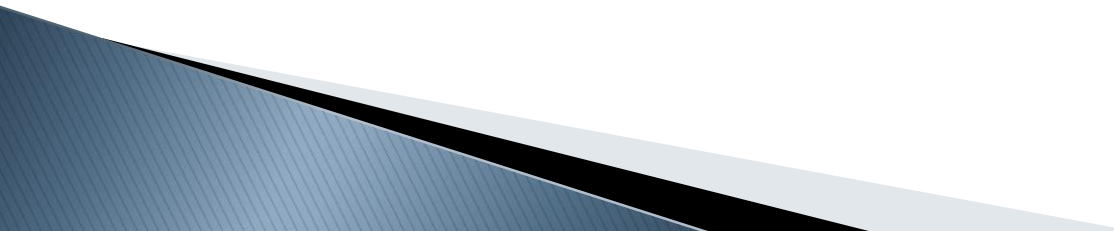
Primeiro momento (revisão)

- ✓ Estruturas de decisão.

Segundo momento (conteúdo)

- ✓ Estruturas de repetição
 - ✓ for
 - ✓ while
 - ✓ do-while

Terceiro momento (síntese)

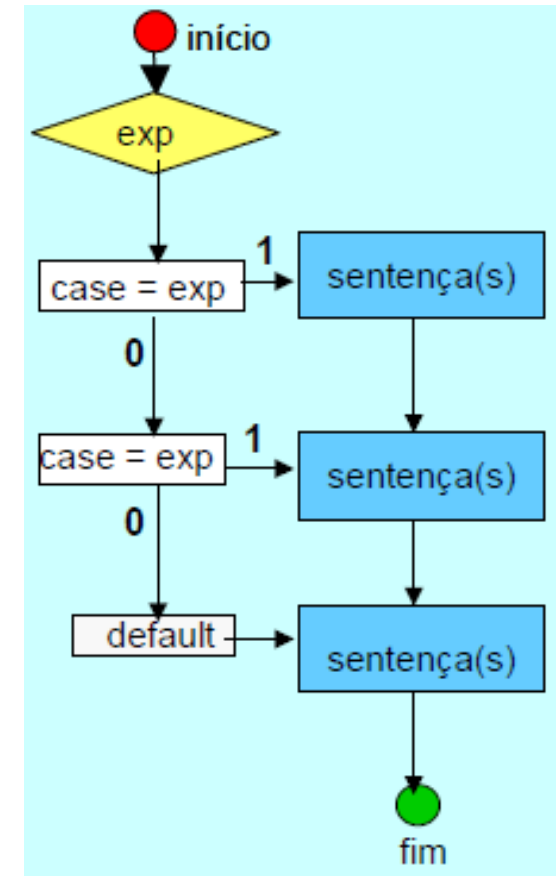
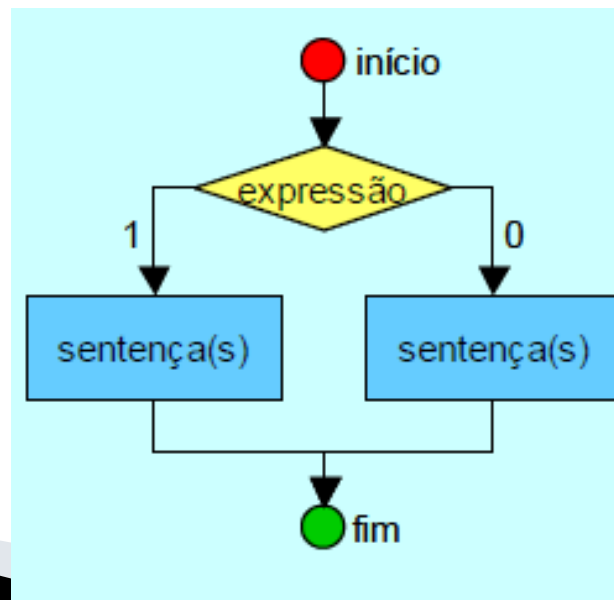
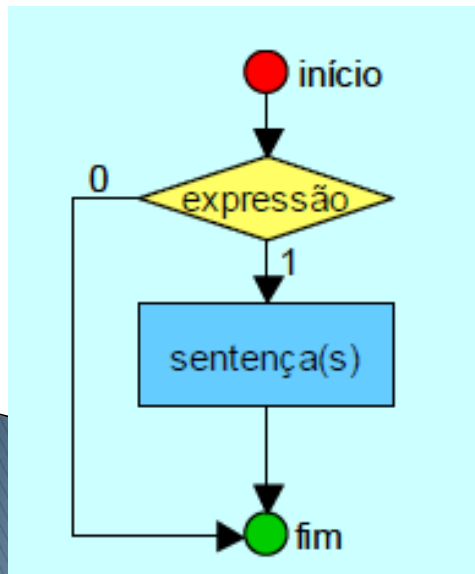
- ✓ Retome pontos importantes da aula
- 

1. Primeiro momento: revisão

Qual a importância das estruturas de decisão em um programa?

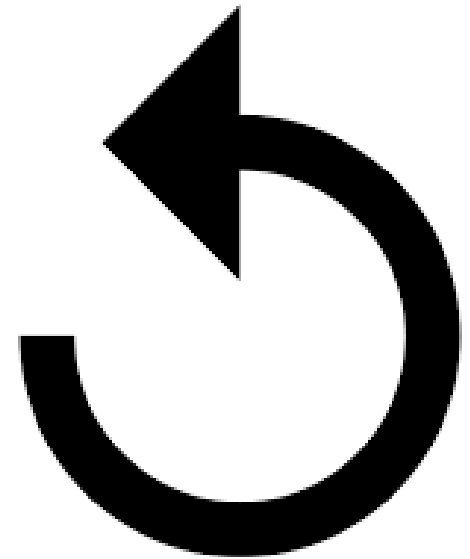
Qual a sintaxe do comando if?

Qual a sintaxe do comando switch-case?



2. Segundo momento

- ▶ Estruturas de repetição



2. Segundo momento

- ▶ Estruturas de repetição
 - for

2. Segundo momento: motivação

Um arquivo histórico municipal guarda informações públicas ou privadas de domínio público. Como exemplo estão: os dados da folha de pagamento da prefeitura e da câmara municipal; os dados dos impostos dos contribuintes; os dados de serviços públicos prestados a todas as propriedades do município, como água e esgoto; os dados dos imigrantes e seus antepassados; etc.

Acontece que a demanda por uso dos dados armazenados é enorme e muitas vezes, para uma única pesquisa, pode-se levar dias até a obtenção de uma resposta.

O Departamento de água de um determinado município precisa provar que efetuou a compra de uma grande área dentro da cidade, na qual está construída uma de suas estações de tratamento de água para distribuição por meio de suas adutoras.

A compra foi efetuada na década de 50 do século passado, e existem dezenas de milhares de caixas no arquivo municipal para pesquisa aos documentos comprobatórios.

Como proceder?

2. Segundo momento: motivação

A Maratona de Programação é uma competição entre times.

Cada equipe deve submeter um a um os programas que consideram corretos.

O critério primordial de classificação é o número de problemas resolvidos com status “Y” de aceito. Em caso de empate por número de problemas resolvidos, fica mais bem classificada aquela com menor Tempo Relativo (TR), incluindo aí, punição de 20min para cada submissão incorreta.

O Professor Antero estava pensando se o resultado da última maratona de programação da FHO seria diferente se não houvesse a punição acima estabelecida.

Então, o cálculo do TR para a seguinte forma: cada vez que a equipe apresentar uma solução com status Y será somado ao seu TR o intervalo de tempo transcorrido desde o início da competição até o momento em que o programa foi submetido.

O prof. Antero está com problema em ver a pontuação de cada equipe.

Como ajudá-lo?

3. Estrutura de repetição (loop)

Um loop é um recurso usado para repetir um comando ou bloco de comandos, tantas vezes quantas forem necessárias, para a resolução de um problema específico dentro do algoritmo.

Cada execução do loop é conhecida como laço ou laçada .

Podem ser classificados como:

- Laços contados : comando **for**.
- Laços condicionais: comandos while e do-while (próxima aula).

4. Comando for

O comando for é classificado como laço contado, seu uso é adequado quando:

- se tem conhecimento prévio de quantas vezes o laço será executado
- é possível calcular o número de vezes de execução do laço

4. Comando for

Sintaxe:

```
for (inicialização; condição; incremento) {  
    lista de comandos;  
}
```

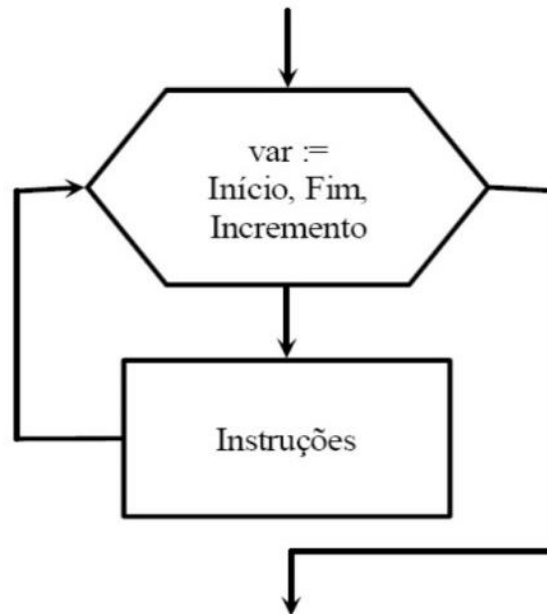
- A inicialização possibilita atribuir valores iniciais para variáveis de controle, incluindo o contador de laçadas.
- “Condição” é executada antes de entrar na laçada. Nela está uma expressão condicional que indica permanecer ou encerrar o loop. Quando a expressão for falsa, o loop é terminado.
- Incremento permite, a cada laçada, alterar o valor de variáveis, incluindo o contador de laçadas. Após o incremento a condição é reavaliada.

4. Comando for

Sintaxe:

```
for (inicialização; condição; incremento) {  
    comandos;  
}
```

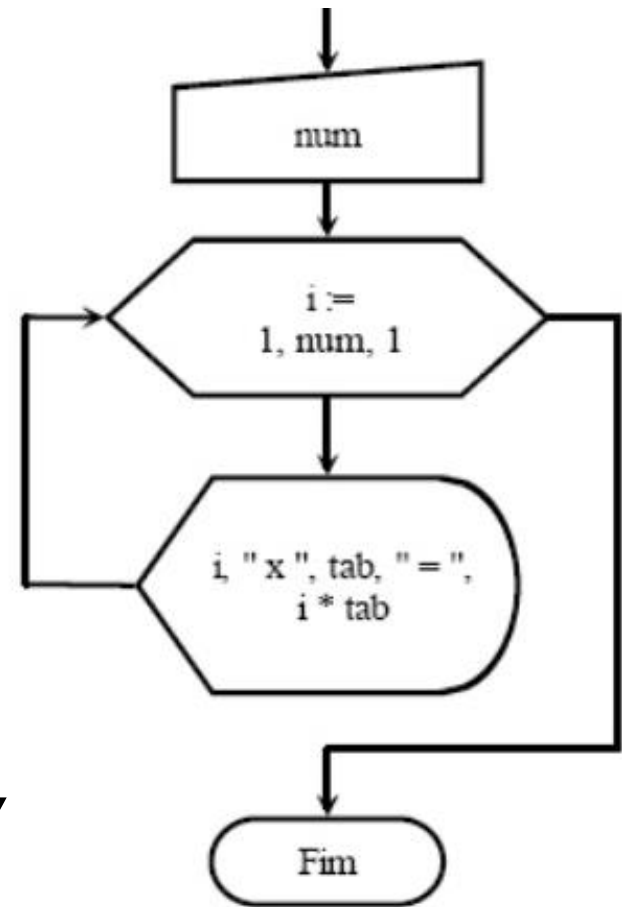
Fluxograma:



4. Comando for

Exemplo: Programa que imprime a tabuada de um determinado número.

```
int main () {  
    int cont, num;  
    printf("Entre com um número: ");  
  
    scanf("%d", &num);  
  
    for (cont=1; cont<=10; cont+=3) {  
        printf ("\n%d X %d = %d", cont,  
num, num*cont);  
    }  
}
```



5. Exercícios

**Vamos
Programar!**



5. Exercícios

- 1) Um conhecido professor, no seu tempo de juventude, conheceu o chinês Ding Ping Lee.

Passados muitos anos, o professor relembrou de uma brincadeira que o chinês criou, usando seu próprio nome.

A brincadeira consiste em ir falando em voz alta, números de 1 a 100. Sempre que encontrar um múltiplo de 3, substituir o número pela palavra Ding. Sempre que for um múltiplo de 5 usar a palavra Ping. Ao término, usar o nome inteiro.

Exemplo:

1, 2, Ding, 4, Ping, Ding, 7, 8, ..., 97, 98, Ding, Ping, Ding Ping Lee.

Desenvolver um programa em Linguagem C, que mostre, na tela, uma saída de dados como a da brincadeira apresentada.

5. Exercícios

2) A regra do impedimento no futebol, em muitos momentos, é difícil de ser aplicada pelo bandeirinha. Ela funciona dada as posições de três jogadores: “L” o jogador atacante que lança a bola; “R” o jogador atacante que recebe a bola; e “D” o último jogador defensor.

E a regra vale somente se o jogador R está no seu campo de ataque; se o jogador R está no seu campo de defesa ou na linha divisória do meio campo, ele não está em impedimento. Neste problema o campo tem 100 metros de comprimento.

Dadas as posições desses três jogadores, no momento exato do lançamento, haverá impedimento se e somente se a seguinte condição for verdadeira:

$(R > 50) \text{ e } (L < R) \text{ e } (R > D)$

Tarefa

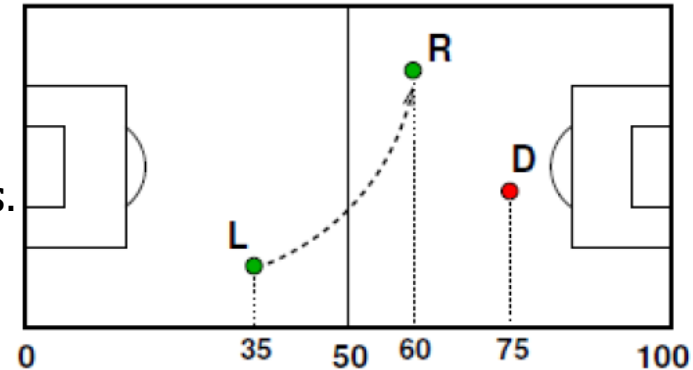
Sua tarefa é produzir um programa que processe vários lances de impedimento. O programa deve apenas determinar, dadas as três posições L, R e D, se há ou não impedimento, implementando exatamente a condição acima. A figura abaixo mostra um exemplo onde não há impedimento.

5. Exercícios

Entrada

A entrada de dados é composta por várias linhas. Na primeira linha tem um número inteiro (N) que representa a quantidade de lances a serem processados.

As N linhas seguintes contém três inteiros, respectivamente as posições dos jogadores L; R e D, sendo $0 \leq L, R, D \leq 100$.



Saída

O programa deve produzir uma linha para cada lance processado, contendo um único caractere, que deve ser “S” caso haja impedimento, ou “N” caso contrário.

Exemplo de Entrada

```
3
35 60 75
55 68 67
66 80 80
```

Exemplo de Saída

```
N
S
N
```

5. Exercícios

3) Tarefa URI1143 Ao Quadrado e ao cubo

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1143>

4) Tarefa URI1079 Médias ponderadas

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1079>

5) URI1060 Números Positivos

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1160>



5. Terceiro momento: síntese

- As estruturas de repetição possibilitam executar repetidas vezes um determinado trecho de código.
 - Esse processo é chamado de loop.
 - Na linguagem C, um loop pode ser escrito pelos comandos “for”, “while” e “do while”.
 - O comando “for” possibilita contar quantas laçadas serão executadas.
- 