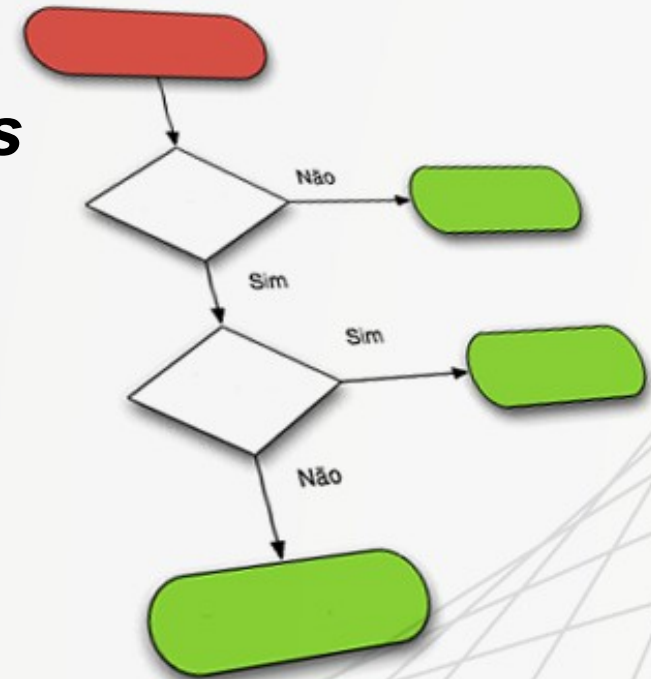


Introdução à Programação

Estrutura e Elaboração de Algoritmos
Variável, entrada e saída
e dados, expressões

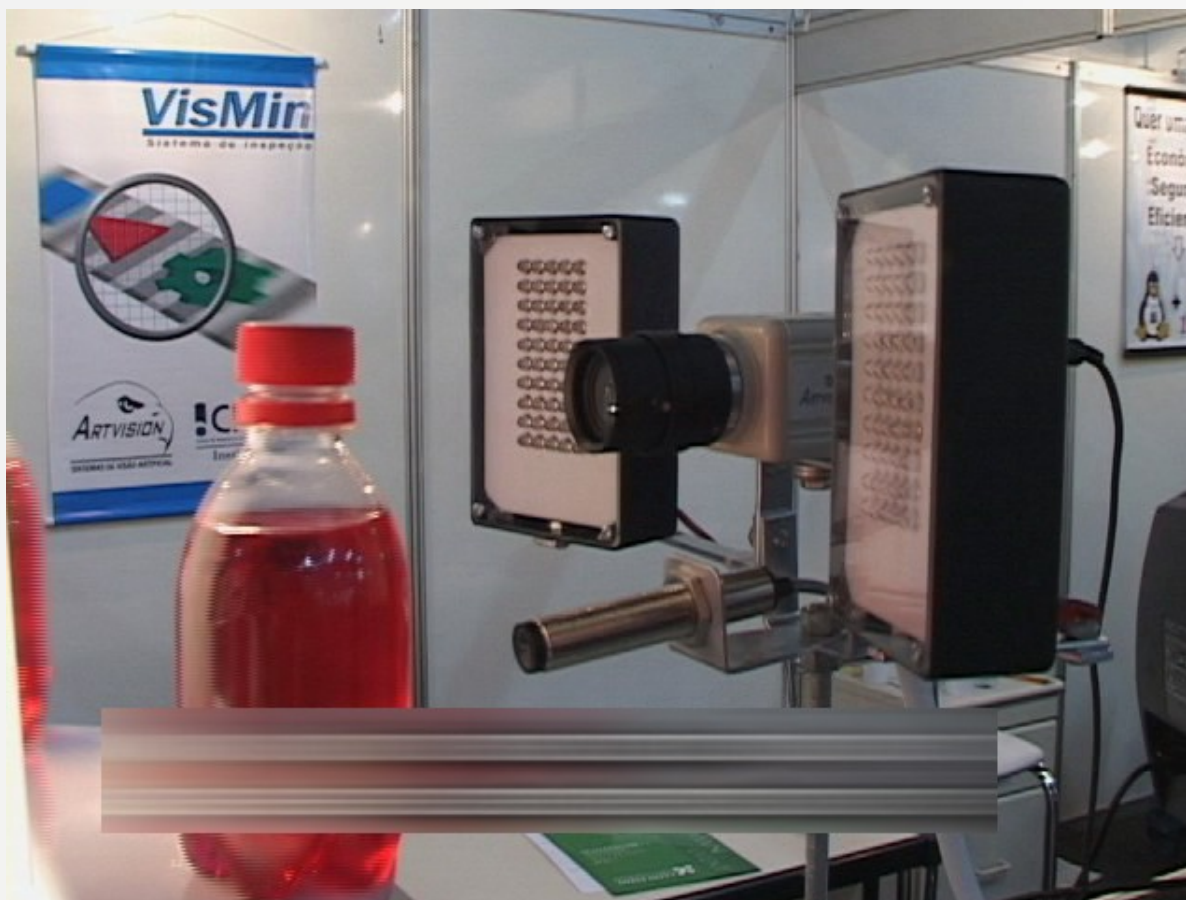


Professor: Saulo Vargas

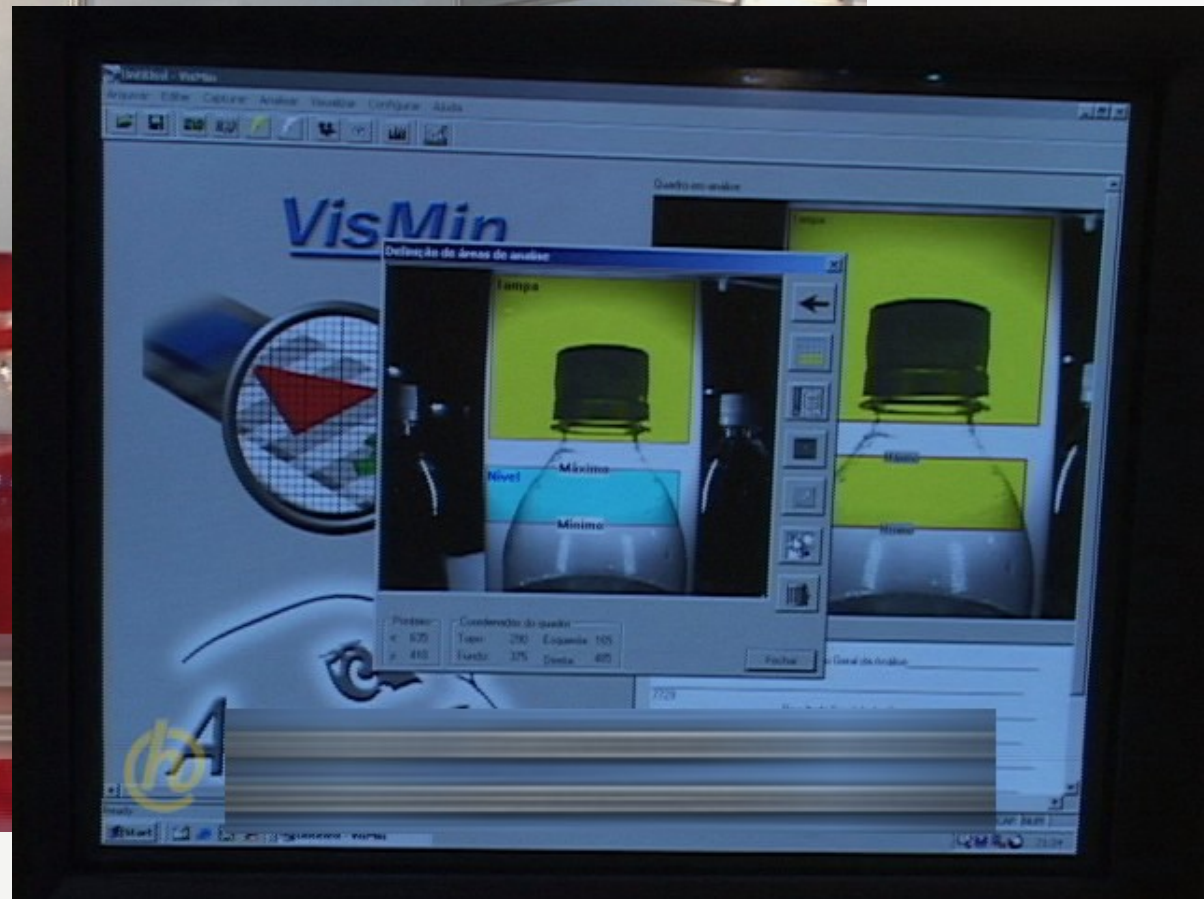
- Plano de Aula
 - Estrutura e Elaboração de Algoritmos
 - Revisão da Aula Anterior
 - Introdução a comandos de entrada e saída
 - Introdução à Variáveis
 - Expressões
 - Exercícios

Curiosidades e Aplicações da Informática

- Inspeção Automatizada de Produtos



- Inspeção Automatizada de Produtos



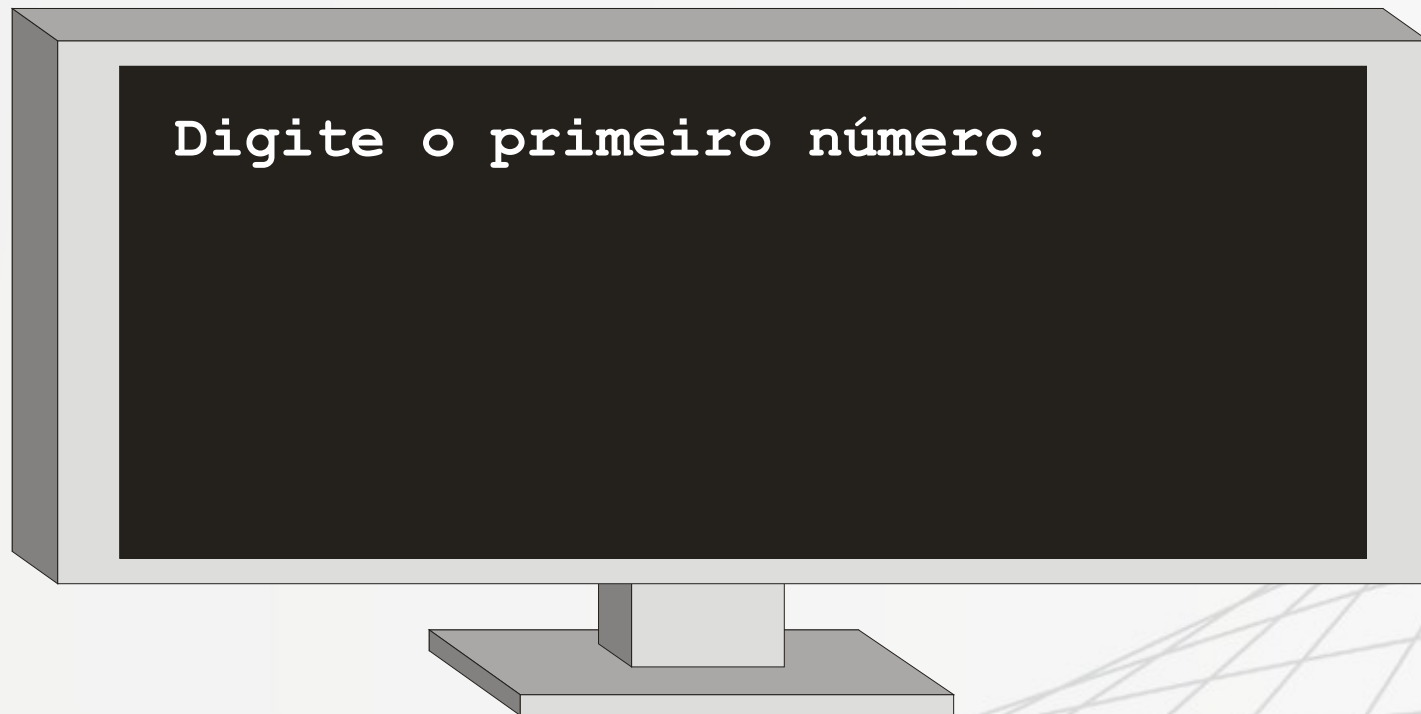
Revisão

Revisão

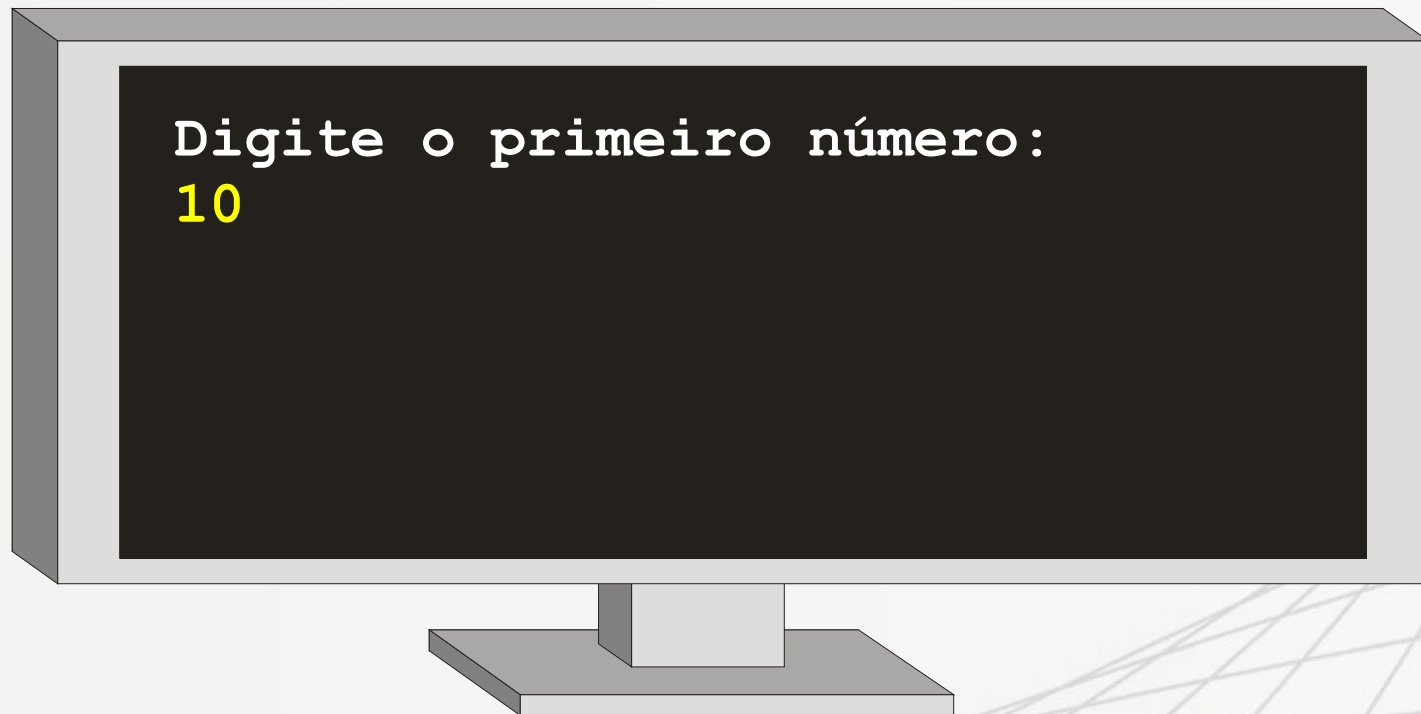
- *Representação de algoritmos*
- *Passos para criação de algoritmos*

Início do Conteúdo

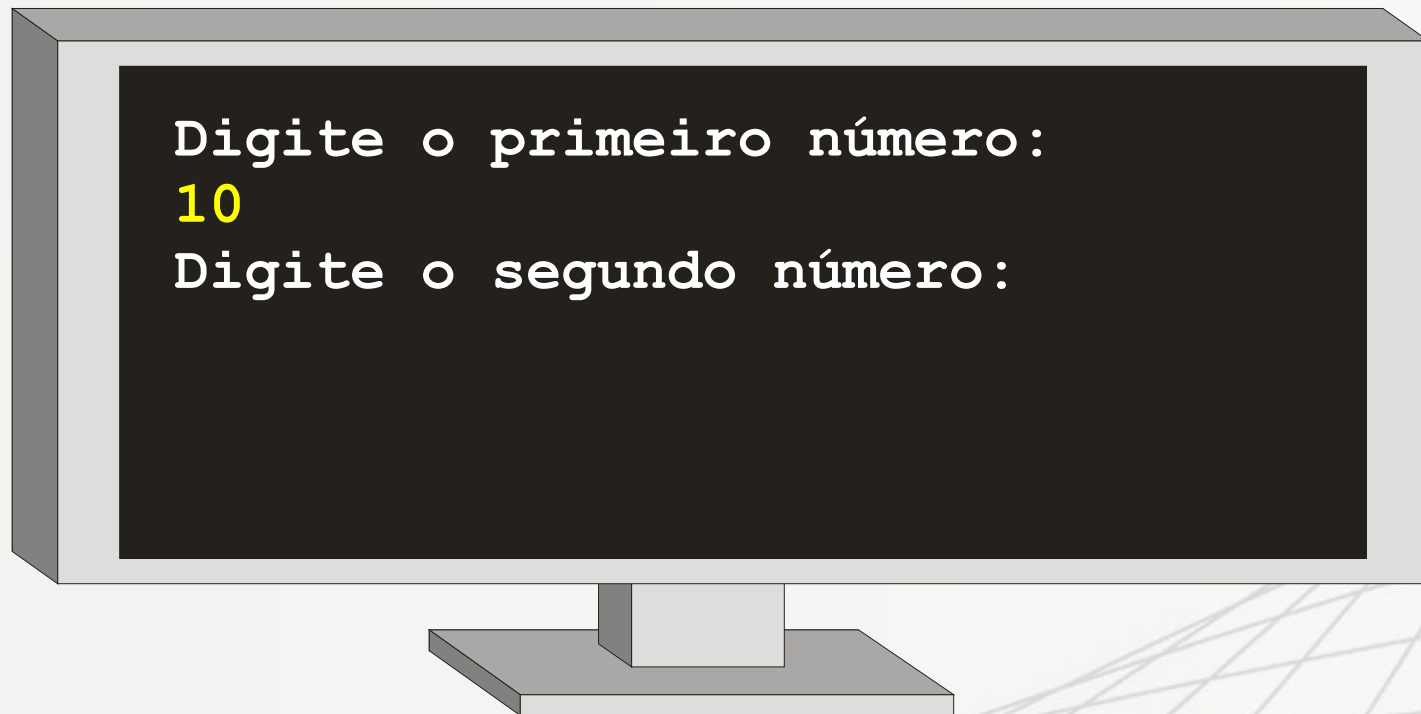
- Considere o seguinte programa



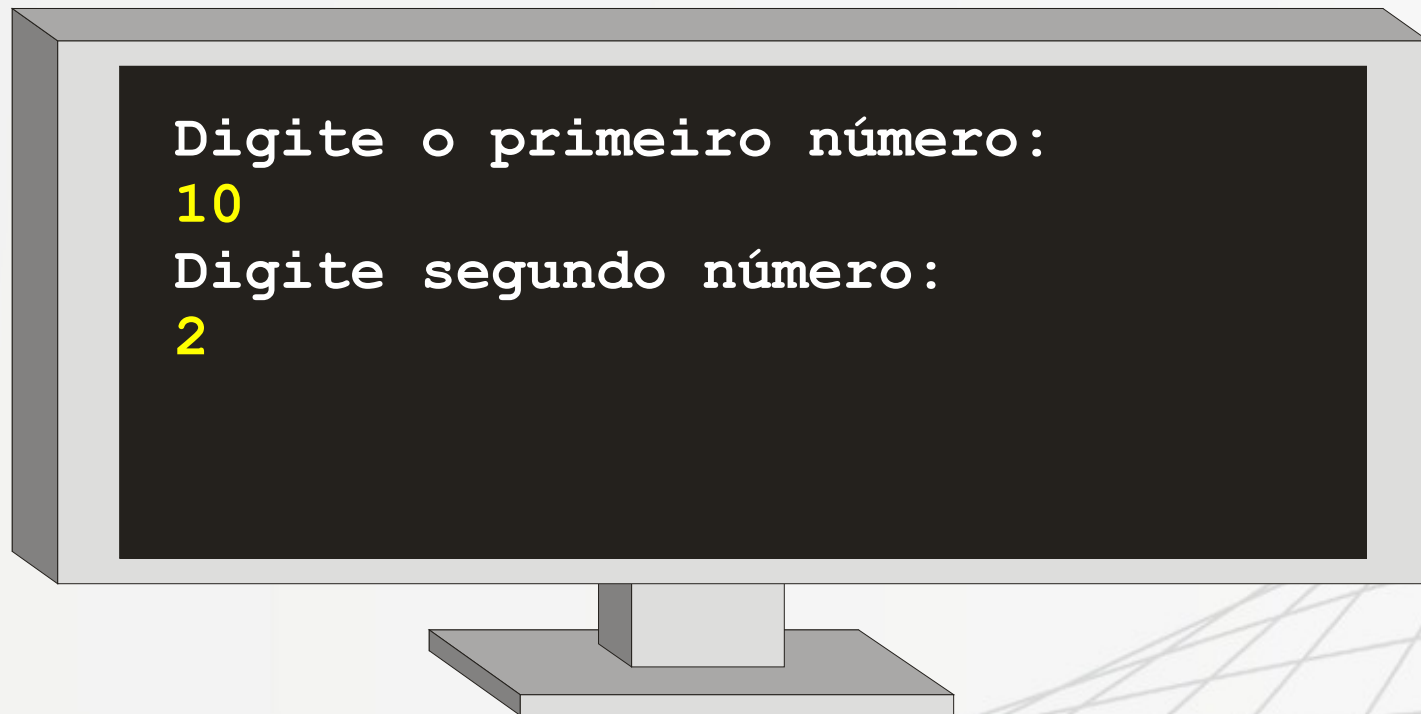
- Considere o seguinte programa



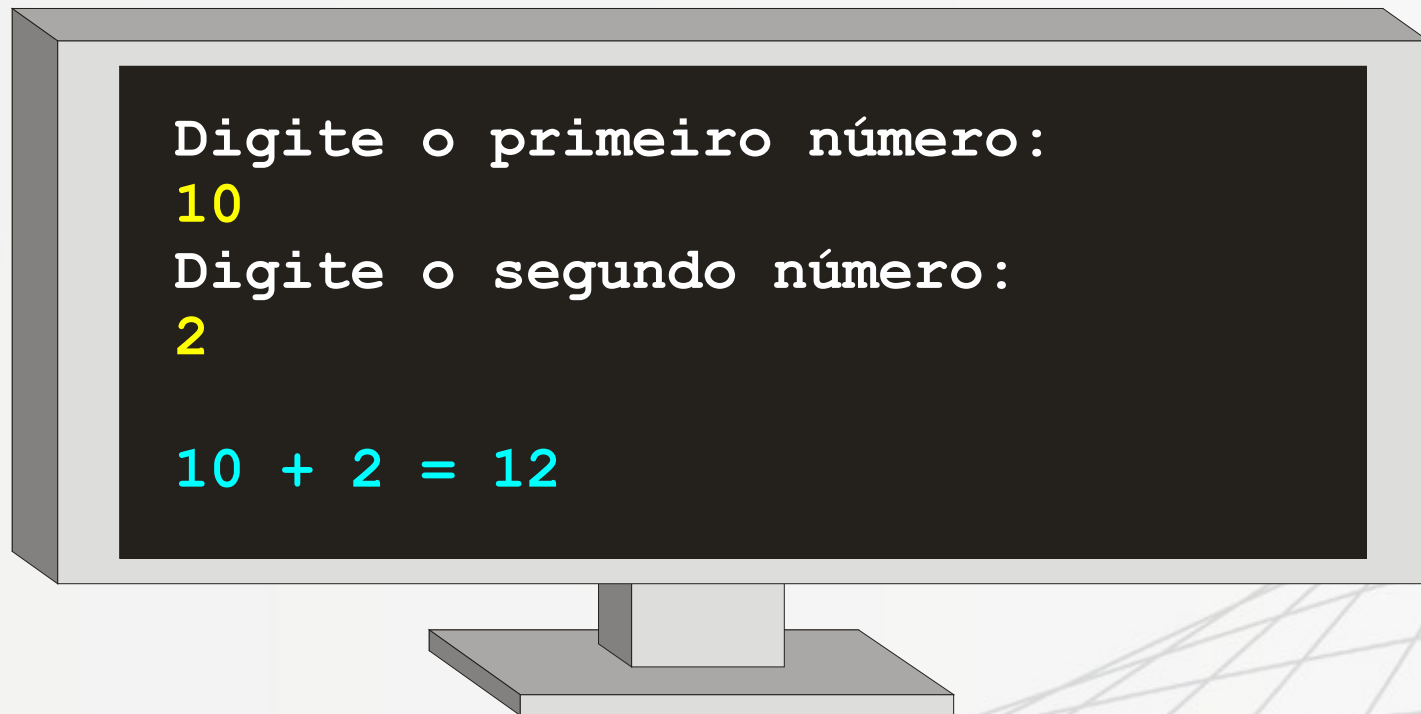
- Considere o seguinte programa



- Considere o seguinte programa



- Considere o seguinte programa



- Imagine:
 - Como é apresentada uma mensagem na tela?
 - Como é realizada a leitura do teclado?
 - Para onde vão os valores lidos pelo teclado?

Variáveis

- **Variável:** Representação de uma posição na memória, onde pode ser armazenado um dado
- Uma **Variável** é referenciada por um nome (identificador) e só pode armazenar um único dado por vez
- Exemplo:
 - **nome** aponta para a posição linha 2, coluna 2
 - **idade** aponta para a posição linha 4, coluna 1

	João	
26		

Variáveis – Nome ou Identificador

- A criação do nome da variável segue as seguintes regras:
 - O primeiro caracter não pode ser um número;
 - Usar letras minúsculas ou maiúsculas (a-z, A-Z), números ou o símbolo _;
 - C é uma linguagem *case sensitive* (NUM ≠ Num ≠ num);
 - O nome não pode ser uma das palavras reservadas da linguagem (float, int, break, if, while, for,...);
 - Exemplos:

Correto

idade
velocidade_x
salBruto

Incorreto

1numero
velocidade...x
sal&Bruto

Variáveis – Tipos de dados

- Uma **Variável** só pode assumir valores de uma mesmo tipo. Neste momento vamos nos concentrar nos seguintes tipos de dados:
 - **int** (números inteiros: ..., -2, -1, 0, 1, 2, ...) 2 bytes
 - **float** (números reais: 5.1, 3, 4.333, ...) 4 bytes
 - **char** (caracteres: "m", "x", "%") 1 byte

Variáveis – Tipos de dados

- O tipo de uma variável é definido através de uma declaração feita no início do algoritmo;
- O tipo da variável não muda durante a execução;
- 4 tipos básicos de dados em C:

Tipo	Espaço	Faixa mínima
char	1 byte	Valores tabela ASCII
int	2 byte	-32.767 a 32.767
float	4 byte	-3.4E-38 a 3.4E+38
double	8 byte	-1.7E-308 a 1.7E+308

Variáveis – Tipos de dados

- O comando para declaração é:
 <**tipo de dado**> <lista de identificadores>;
- Exemplos:
 int idade;
 float peso, altura;
 char sexo;

Variáveis – Tipos de dados

- Exercício

1) Indique o tipo de variável que pode ser adotado para cada situação a seguir (considere apenas: int, float ou char):

- Quantidade de membros por família
- Renda familiar
- Alternativa escolhida em questões múltipla escolha(A-E)
- Preço do litro da gasolina
- Altura de um pessoa
- Quantidade de mulheres inscritas
- Operação aritmética escolhida(-, -, *, /)
- Número de analfabetos em um país

Variáveis – Tipos de dados

- Quantidade de recém-nascidos
- Classe social (A, B, C, D)
- Distância percorrida por um carro
- Quantidade de litros abastecidos
- Alternativa escolhida em questões de V ou F
- Número de passageiros transportados por dia
- Temperatura média diária
- Valor arrecadado em impostos
- Número de leitos hospitalares
- Velocidade de um carro
- Área de um retângulo

Variáveis – Alteração do valor

- O valor de uma **Variável** pode alterar durante a execução do algoritmo.
- Esta alteração pode se dar por:
 - *Comando de Entrada de dados*
 - *Comando de Atribuição*

Variáveis – Alteração do valor – Comando de Atribuição

- *Atribuição: Através deste comando podemos armazenar valores constante, valores de outras variáveis ou mesmo o resultado de expressões.*
 - *Formato do Comando:*
`<identificador> = <valor a ser armazenado>;`
 - *Exemplos*
`num1 = 50;`
`produto = 10 * 5;`
`total = quantidade * valor;`
`num1 = 30;`

Comandos de Entrada e Saída de Dados

- ***Comando de Entrada:*** usado para dados do teclado

- *Formato do comando*

```
scanf("formato_do_identificador", &<identificador>);
```

- *Exemplos*

```
int idade;
```

```
scanf("%d", &idade);
```

```
float peso, altura;
```

```
scanf("%f %f", &peso, &altura);
```

Comandos de Entrada e Saída de Dados

- **Análise do comando de entrada *scanf()***

`scanf("formato_do_identificador", &<identificador>);`

- `scanf`: palavra-chave, representa o comando de entrada pelo teclado
- `"formato_do_identificador"`: %d – int; %f – float; %c – char; %s – string. Deve(m) estar entre aspas
- `identificador`: nome da variável. Deve ser precedido do operador `&`
- a linha de comando possui no final um ponto e vírgula `(;)`

Comandos de Entrada e Saída de Dados

- **Comando de Saída:** usado para mostrar o resultado do processamento e de mensagens no monitor.
 - Formato do comando

`printf("formato_da_expressão", expressão);`

Exemplos

```
int idade = 26;  
printf("%d", idade);
```

Saída no monitor

26

```
float peso = 2.459;  
printf("Peso = %f", peso);
```

Peso = 2.459000

```
float peso = 84 altura = 1.83;  
printf("Peso=%f Altura=%f", peso, altura);
```

Peso=84.00000 Altura=1.830000

Comandos de Entrada e Saída de Dados

- **Análise do comando de saída *printf()***

`printf("formato_da_expressão", expressão);`

- `printf`: palavra-chave, representa o comando de saída pelo monitor
- `formato_da_expressão`: %d – int; %f – float; %c – char; %s – string. Deve(m) estar entre aspas.

Aqui também podem aparecer *palavras* que deseja imprimir e *escapes* (\n – quebra linha; \a – sinal sonoro;...) As *palavras* e os *escapes* são passados para a saída na ordem em que foram escritos.

Exemplo: `printf("Peso = %f \n", peso);`

Comandos de Entrada e Saída de Dados

- **Análise do comando de saída *print()***

```
printf("formato_da_expressão", expressão);
```

Além disso, no campo `formato_da_expressão`, pode-se definir número de casas decimais

Exemplo:

```
float imc = 1.63556;
```

```
printf("imc = %.2f \n", imc);
```

Saída no monitor

```
imc = 1.64
```

Comandos de Entrada e Saída de Dados

- **Análise do comando de saída *print()***

`printf("formato_da_expressão", expressão);`

- **expressão**: pode ser nome de uma variável, expressão aritmética/lógica ou cadeia de caracteres.

Exemplos:

`int horas = 15;`

`printf("Horas = %d", horas);`

`printf("Result = %d", 24 - horas);`

`printf("Digite um valor");`

Saída no monitor

Horas = 15

Result = 9

Digite um valor

Comandos de Entrada e Saída de Dados

- Os comandos `scanf()` e `printf()` precisam da biblioteca `stdio.h`.
(`#include <stdio.h>`)

Estrutura básica

<diretivas do pré-processador>

main(){

 <declarações locais>;

 <instrução 1>;

 <instrução 2>;

 <instrução n>;

}

Estrutura básica

- **Exemplo:** Calcular a média de 4 números

```
//calcula média de 4 números → comentário  
#include <stdio.h> → acrescenta funcionalidades da biblioteca stdio  
  
main() → função principal  
{  
    float n1, n2, n3, n4, media;  
  
    scanf("%f", &n1);  
    scanf("%f", &n2);  
    scanf("%f", &n3);  
    scanf("%f", &n4);  
    media = (n1+n2+n3+n4) / 4;  
    printf("Média = %.2f \n", media);  
}
```

→ instruções

Expressões

- Utilizadas para obter novos valores a partir dos valores de variáveis e constantes
- As expressões são formadas utilizando constantes, variáveis, operadores e funções
- Operadores: são símbolos usados para efetuar algum cálculo sobre as variáveis e constantes
- Funções: também são utilizados para realizar algum cálculo, mas normalmente mais complexos. São como pequenos programas, que tendo valores como entrada geram uma resposta.

Expressões

- Tipos de Expressões
 - Expressões Aritméticas
 - Expressões Literais
 - Expressões Relacionais
 - Expressões Lógicas

Expressões Aritméticas

- Utilizadas com constantes e variáveis numéricas

Operadores	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão (resultado da divisão de números inteiros x e y é a parte inteira de x/y) Ex. $7 / 2 = 3$
%	Resto da divisão inteira de x por y (somente números inteiros) Ex. $5 \% 3 = 2$

Expressões Aritméticas

- Utilizadas com constantes e variáveis numéricas

Funções	Descrição
pow(x,y)	x elevado a y - Ex. pow(2,3) = 8
sqrt(x)	Raiz quadrada de x
tan(x)	Tangente de x
cos(x)	Cosseno de x
sin(x)	Seno de x
log10(x)	Logaritmo decimal - Ex. log10(100) = 2
M_PI	3.14159...

- Precisam da biblioteca math.h (`#include <math.h>`)

Prioridade no cálculo de expressões

1º lugar	Parênteses mais internos	
2º lugar	Funções	
3º lugar	Operadores aritméticos	1º lugar: ^ 2º lugar: *, / 3º lugar: +, -

Ex. $\text{sqrt}(30+6)*2 = 12$

Observação: Havendo empate em uma expressão, deve-se fazer primeiro o cálculo da esquerda

Ex. $10/5*2 = 4$

Expressões Aritméticas

- Transformando expressões do formato matemático para o formato utilizado em C:

$$\frac{1}{A} \sqrt{30 - y} \longrightarrow 1 / A * \text{sqrt}(30 - y)$$

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a} \longrightarrow (-b + \text{sqrt}(\text{pow}(b, 2) - 4 * a * c) / (2 * a)$$

Escrevendo o primeiro programa

A partir de agora vamos escrever nossos algoritmos usando a estrutura básica e os comandos da linguagem C

<diretivas do pré-processador>

main()

{

 <declarações locais>;

 <instrução 1>;

 <instrução 2>;

 <instrução n>;

}

Escrevendo o primeiro programa

- Exemplo 1: Escreva um algoritmo que receba como entrada uma dada temperatura em graus Celsius e mostre a temperatura correspondente em graus Fahrenheit

Conversão: $\text{Fahrenheit} = 1.8 * \text{Celsius} + 32$

Escrevendo o primeiro programa

- Exemplo1: Escreva um algoritmo que receba como entrada uma dada temperatura em graus Celsius e mostre a temperatura correspondente em graus Fahrenheit

Conversão: Fahrenheit = $1.8 * \text{Celsius} + 32$

Resolução:

- 1 - Entendeu o problema?
- 2 – Entrada: Temperatura em graus Celsius
- 3 – Saída: Temperatura em graus Fahrenheit
- 4 – Transformar entrada em saída:

Fahrenheit = $1.8 * \text{Celsius} + 32$

- 5 – Escrever o algoritmo

Escrevendo o primeiro programa

- Exemplo 1: Escreva um algoritmo que receba como entrada uma dada temperatura em graus Celsius e mostre a temperatura correspondente em graus Fahrenheit

```
#include <stdio.h>

main()
{
    float tempFah, tempCel;

    scanf("%f", &tempCel);
    tempFah = 1.8*tempCel + 32;
    printf("%f", tempFah);
}
```

Escrevendo o primeiro programa

- Exemplo 1: Escreva um algoritmo que receba como entrada uma dada temperatura em graus Celsius e mostre a temperatura correspondente em graus Fahrenheit

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

```
main()  
{
```

```
    float tempFah, tempCel;
```

```
    printf("Digite temperatura Celsius \n");
```

```
    scanf("%f", &tempCel);
```

```
    tempFah = 1.8*tempCel + 32;
```

```
    printf("%.2f Celsius = %.2f Fahrenheit", tempCel, tempFah);
```

```
}
```

Escrevendo o segundo programa

- Exemplo 2: Escreva um algoritmo que receba como entrada o valor da base e da altura de um triângulo e mostre a o valor da área.

Escrevendo o segundo programa

- Exemplo 2: Escreva um algoritmo que receba como entrada o valor da base e da altura de um triângulo e mostre a o valor da área.

Resolução:

- 1 - Entendeu o problema?
- 2 – Entrada: base e altura do triângulo
- 3 – Saída: área do triângulo
- 4 – Transformar entrada em saída:
$$\text{area} = \text{base} * \text{altura} / 2$$
- 5 – Escrever o algoritmo

Escrevendo o segundo programa

- Exemplo 2: Escreva um algoritmo que receba como entrada o valor da base e da altura de um triângulo e mostre a o valor da área.

```
#include <stdio.h>

main()
{
    float base, altura, area;

    scanf("%f %f", &base, &altura);
    area = base*altura/2;
    printf("Area = %.2f", area);
}
```


RESOLVA A LISTA 1 NO SIGAA

- Referência Bibliográfica

- SCHIIDT, Herbert. C completo e total. 3 ed. São Paulo: Pearson, 1997.
- SENNE, Edson Luiz França. Primeiro curso de programação em C. 3 ed. Florianópolis: Visual Books, 2009
- PIVA JUNIOR, Dilermando, et al. Algoritmos e programação de computadores. Rio de Janeiro: Elsevier, 2012
- Prof. Maurício Edgar Estivanello. Notas de aulas de Introdução à Programação. Disponível em: moodle.gaspar.ifsc.edu.br, 2013.