

Período: 1º Semestre

Disciplina: Algoritmos e Lógica de Programação

Professor: Profa. Dra. Ligia Rodrigues Prete

E-mail: ligia.prete@fatec.sp.gov.br

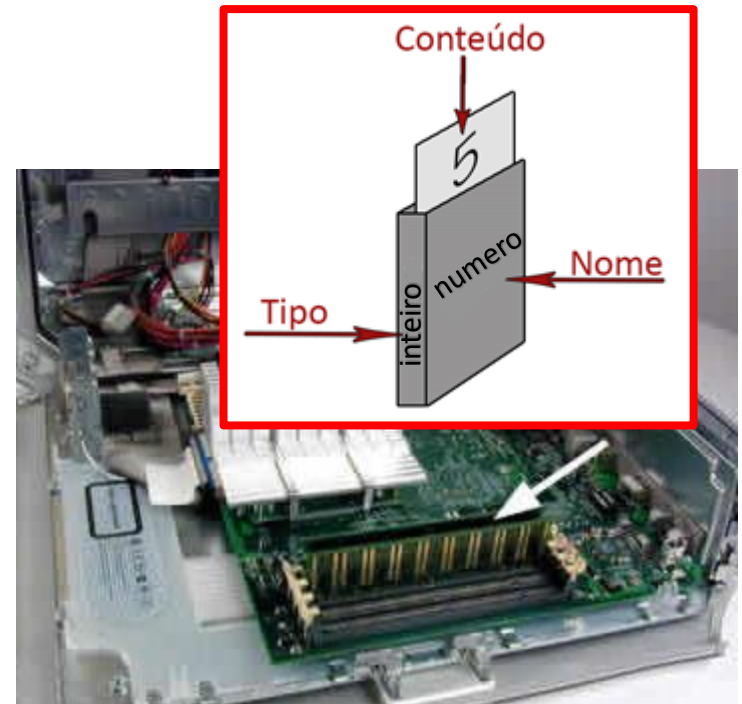
5 – Variável

Variável

A variável representa um valor no algoritmo. Ela possui:

- Nome;
- Conteúdo;
- Tipo de dados;
- Só existe em tempo de execução (memória RAM).

Exemplo:
variável numero: inteiro



Posição da variável na memória RAM do computador

Os valores das variáveis são armazenadas na memória RAM do computador

	variável numero = 5	

Memória RAM

Nome de Variáveis

- O nome de uma variável pode ser constituído por letras do alfabeto (minúsculas ou maiúsculas), dígitos (0 ... 9) e ainda pelo caractere underscore (_);
- O primeiro caractere não pode ser um dígito. Terá que ser uma letra. **Exemplo: nome1;**
- Maiúsculas e minúsculas representam caracteres diferentes, logo variáveis distintas. **Exemplo: num, Num;**
- Uma variável não pode ter por nome uma palavra reservada da própria linguagem C. **Exemplo: printf;**
- Não é aconselhável a utilização de caracteres acentuados no nome das variáveis. **Exemplo: média;**
- O caractere underscore (_) é habitualmente usado para fazer a separação entre palavras. **Exemplo: nome_cli, id_fatura.**

Exemplos de Variáveis

- `int idade;` ✓
- `int Num_Cliente;` ✓
- `float a1b2c3;` ✓
- `float 7a2b3c;` ✗ // incorreto: começou com número
- `char float;` ✗ // incorreto: palavra reservada
- `double vinte%;` ✗ // incorreto: % representa MOD

Atribuição

O operador de atribuição simboliza, basicamente, o armazenamento de um valor ou o resultado de uma expressão. Ele é representado pelo sinal de igual (=).

Exemplo:

$$\mathbf{r} = \mathbf{a} + \mathbf{b}$$

A expressão $\mathbf{a} + \mathbf{b}$ será processada e o seu resultado será armazenado na variável \mathbf{r} .

Comando de Entrada de Dados

- A função para entrada de dados na Linguagem C é o **scanf** (**scan** + **f**formatted – leitura formatada).

Algoritmo

leia ()

Linguagem C

scanf ()

Comando de Saída de Dados

- A função para saída de dados na Linguagem C é o **printf** (**print** + **f**formatted – escrita formatada).

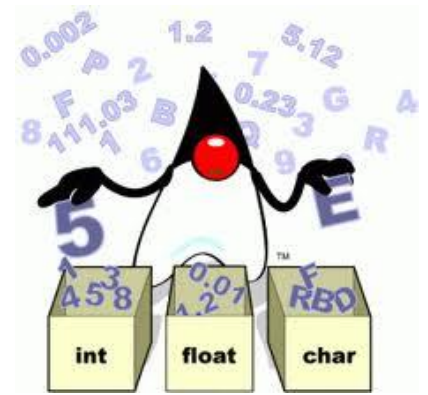
Algoritmo

escreva ()

Linguagem C

printf ()

Tipos de Dados

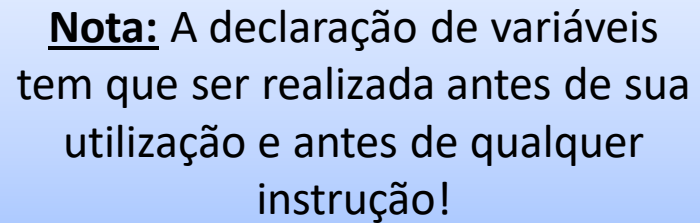


- **Inteiro (int):** armazena somente números inteiros;
Exemplo: -4, 0, 45
- **Real (float e double):** números que possuem a parte fracionária;
Exemplo: -4.50, 78.344, 7.0
- **Caractere (char):** é todo texto colocado entre aspas;
Exemplo: “Olá mundo”, “Algoritmo”, “123”
- **Lógico (bool):** armazena dois valores
Exemplo: verdadeiro ou falso

Declarando Variáveis

- É necessário declarar as variáveis que serão utilizadas durante a execução do programa. Exemplo:

```
char nome;  
int num1, num2;  
float salario;  
double valor;  
bool questao;
```



Nota: A declaração de variáveis tem que ser realizada antes de sua utilização e antes de qualquer instrução!

- Nas linhas acima foram declaradas:
 - Uma variável chamada **nome** do tipo caractere (**char**);
 - Duas variáveis **num1** e **num2** que são do tipo inteiro (**int**);
 - Uma variável **salario** que é do tipo real (**float**);
 - Uma variável **valor** que é do tipo real (**double**);
 - Uma variável **questao** que é do tipo lógico (**bool**). O número “1” representa o resultado verdadeiro e “0” representa o resultado falso.
- Observe quando são declaradas várias variáveis do mesmo tipo, elas devem ser separadas por vírgula.

Formatos de Leitura e Escrita

- A leitura e escrita de variáveis podem ser realizadas através das funções `scanf` (leitura) e `printf` (escrita), utilizando os **formatos** adequados para serem lidos ou escritos.

Algoritmo	Linguagem C	Formato
caractere	<code>char</code>	<code>%c</code>
cadeia	<code>char[]</code>	<code>%s</code>
inteiro	<code>int</code>	<code>%d</code>
real	<code>float</code>	<code>%f</code>
real	<code>double</code>	<code>%f</code>
logico	<code>bool</code>	<code>%d</code>

Tipo Inteiro (int)

- As variáveis declaradas do tipo **inteiro** são utilizadas para armazenar valores que pertencem ao conjunto dos números naturais (sem parte fracionária) positivos e negativos. Exemplo: -345, 0, 2, 115

```
void main()  
{  
  
    int num;  
  
}
```

Exemplo com tipo Inteiro (int)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void main()
5  {
6
7      int num;
8
9      printf("Digite um numero: ");
10
11
12     scanf("%d", &num);
13     printf("O numero digitado foi %d \n", num);
14     system("pause");
15 }
16
```

```
Digite um numero: 10
O numero digitado foi 10
Pressione qualquer tecla para continuar. . .
```

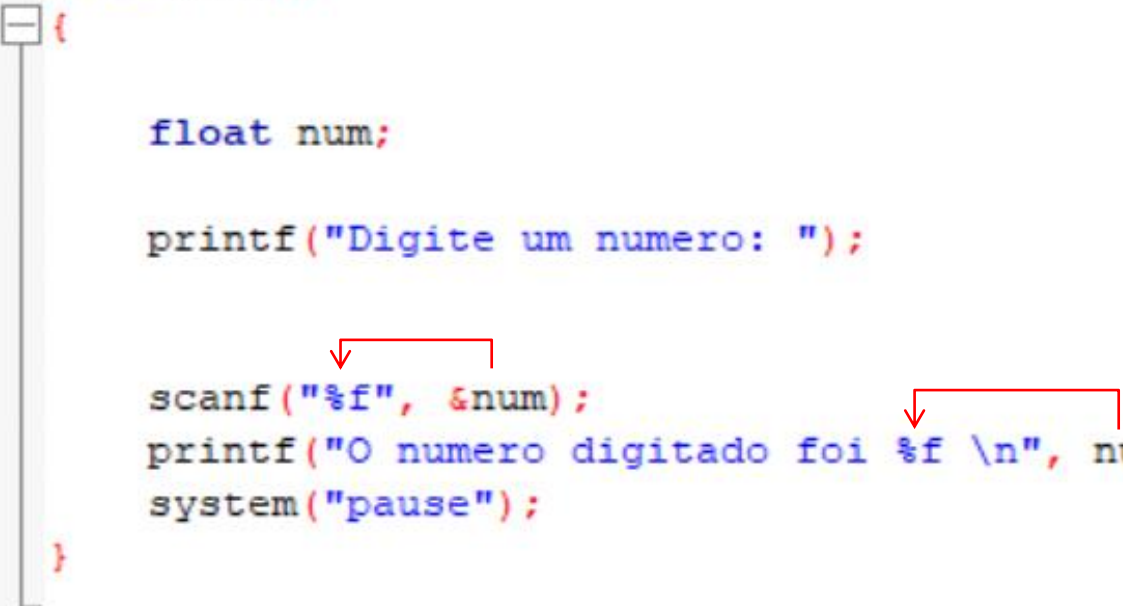
Tipo Real (float e double)

- As variáveis declaradas do tipo **float** ou **double** são utilizadas para armazenar valores numéricos com parte fracionária. A diferença entre eles é o número de bytes que reserva para armazenar o valor. Exemplo: -345.8, 0, 2.5, 115.3

```
void main()  
{  
  
    float num;  
  
}
```

Exemplo com tipo Real (float e double)

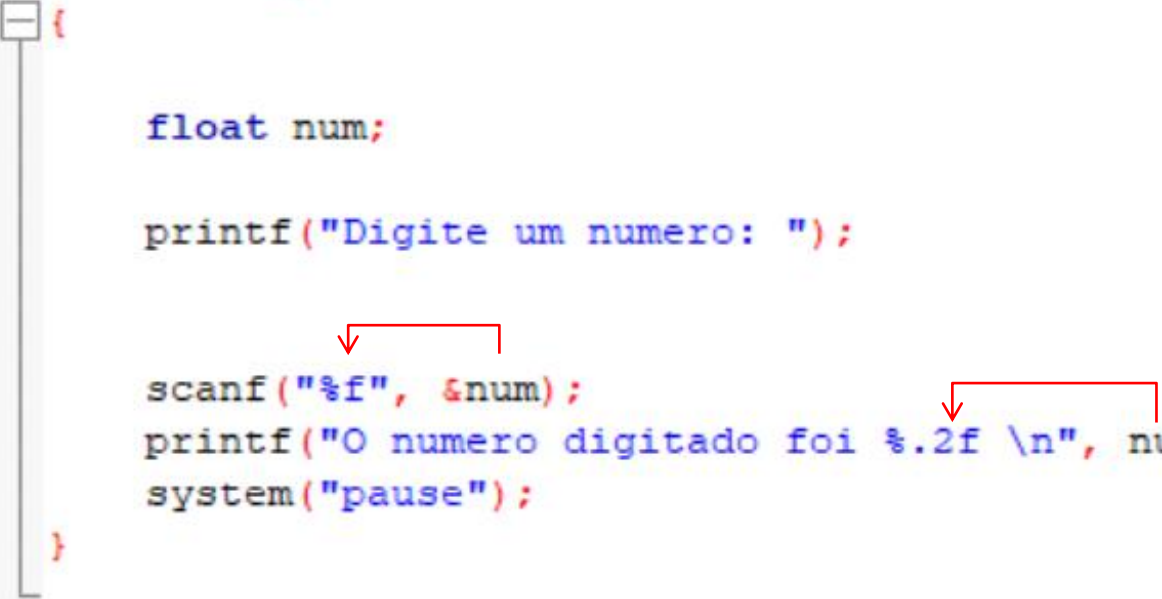
```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void main()
5  {
6
7      float num;
8
9      printf("Digite um numero: ");
10
11
12     scanf("%f", &num);
13     printf("O numero digitado foi %f \n", num);
14     system("pause");
15 }
16
```



```
Digite um numero: 10
O numero digitado foi 10.000000
Pressione qualquer tecla para continuar. . .
```

Formatando tipo Real (float e double)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void main()
5  {
6
7      float num;
8
9      printf("Digite um numero: ");
10
11     scanf("%f", &num);
12     printf("O numero digitado foi %.2f \n", num);
13     system("pause");
14 }
15
16
```



```
Digite um numero: 10
O numero digitado foi 10.00
Pressione qualquer tecla para continuar. . . _
```

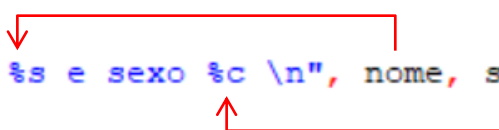

Tipo Caractere (char)

- O tipo **char** permite armazenar um único caractere numa variável desse tipo. Exemplo: **char sexo**
- Já a variável do tipo **cadeia**, popularmente conhecida como string, suporta bem mais de um caractere. Sua declaração é feita da seguinte forma: **char nome[50]**

```
void main()  
{  
  
    char nome[50];  
    char sexo;  
  
}
```

Exemplo com tipo Caractere (char)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void main() {
5
6      char nome[50];
7      char sexo;
8
9      printf("Digite o seu nome: ");
10     gets(nome);
11     printf("Digite o seu sexo: ");
12     scanf("%c", &sexo);
13     printf("\nO nome digitado foi %s e sexo %c \n", nome, sexo);
14     system("pause");
15 }
16
```



```
Digite o seu nome: Ligia Prete
Digite o seu sexo: f

O nome digitado foi Ligia Prete e sexo f
Pressione qualquer tecla para continuar. . .
```

Tipo Lógico (bool)

- Na linguagem C o tipo **bool** apresenta resultados:
 - Verdadeiro com o número “1”
 - Falso com o número “0” .
- Antes de usá-lo deve-se incluir a diretiva:

```
#include <stdbool.h>
```

```
#include <stdbool.h>

void main()
{

    bool v, f;

}
```

Exemplo com tipo Lógico (bool)

```
1  #include <stdbool.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  void main()
6  {
7      bool v, f;
8
9      v = true;
10     f = false;
11
12     printf("O melhor time do Brasil e o Corinthians? %d\n", f);
13     printf("2 + 2 = 4? %d\n", v);
14     system("pause");
15 }
16
```

```
O melhor time do Brasil e o Corinthians? 0
2 + 2 = 4? 1
Pressione qualquer tecla para continuar. . .
```

0 – falso
1 – verdadeiro