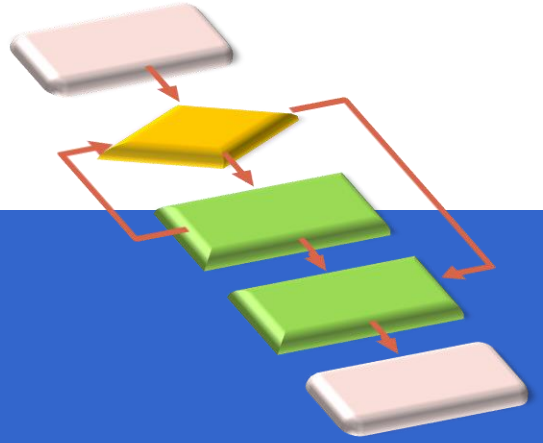
INSTITUTO DE COMPUTAÇÃO

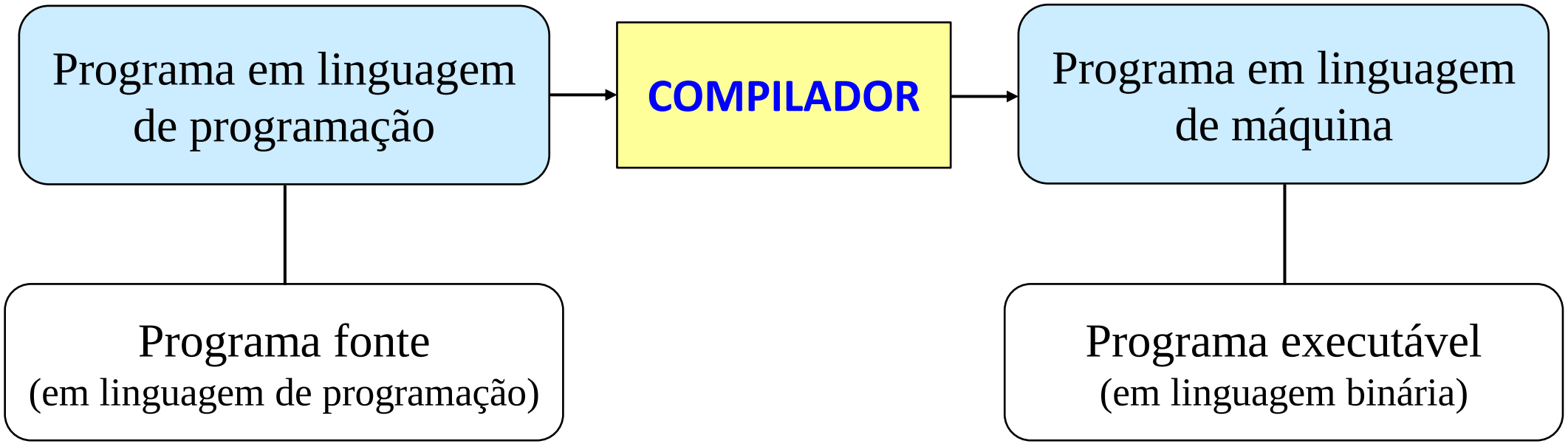


Algoritmos II

CONCEITOS INTRODUTÓRIOS
Tipos simples de dados e declarações

Prof.ª Vanessa de Oliveira Campos

Processo de compilação de programa-fonte



Estrutura de um Algoritmo

- Um algoritmo em pseudolinguagem tem a seguinte estrutura:

```
Algoritmo "< nome do algoritmo >"
Var
        // Seção de Declarações das variáveis
Inicio
        // Seção de comandos
Fimalgoritmo
```

Tipos básicos de dados

- Nas linguagens de programação, os dados são guardados em variáveis (que representam posições de memória).
- Cada uma dessas variáveis está relacionada a um tipo de dado:
  - Numérico (inteiro ou real);
  - Literal (palavra);
  - Caractere;
  - Lógico (Verdadeiro ou Falso);
  - Endereço de memória.
- Também se pode criar novos tipos através de registros (struct) e typedef.

Tipos básicos de dados na Liguagem C

Tipo	Descrição	Tamanho	Intervalo	Exemplos
int	Inteiros positivos ou negativos, sem parte fracionária.	4 bytes	-214783648 a 214783647	-23 98 0 -1346
float	Números reais positivos ou negativos, com menor precisão na parte decimal.	4 bytes	-3.4e38 a +3.4e38	22.54 4.523 78.375 -342.35
double	Números reais positivos ou negativos, com maior precisão na parte decimal.	8 bytes	-1.7e308 a +1.7e308	22.00254 4.52334213 -42.35

Tipos básicos de dados na Liguagem C

Tipo	Descrição	Tamanho	Caracteres armazenados	Exemplos
char	Um único caractere entre aspas simples. Podendo conter letras maiúsculas, ou minúsculas, ou números ou caracteres especiais.	1 byte	caracteres do código ASCII	‘a’ ‘G’ ‘1’ ‘@’
char [ ]	Um ou mais caracteres entre aspas duplas.	4 bytes	caracteres do código ASCII	“Rua A” “3.25” “78060-600”

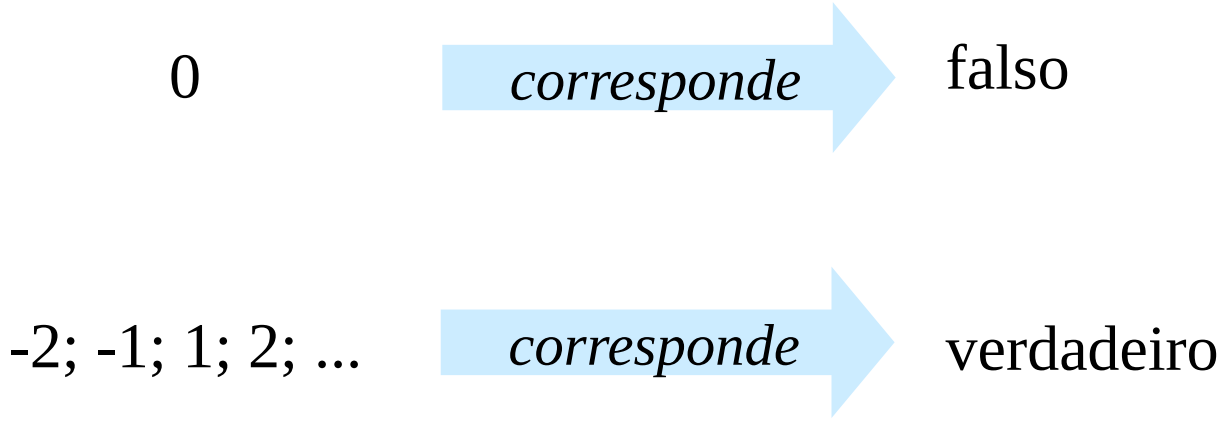
Números – Linguagem C

- Em números fracionários, é utilizado o ponto decimal em lugar da vírgula.
- Notação exponencial - Exemplos:

Números em C	Correspondente
-1.8e2	$-1{,}8 \times 10^{2}$
7.5E-3	$7{,}5 \times 10^{-3}$
0.236E-5	$0{,}236 \times 10^{-5}$

Valores Lógicos – Linguagem C

- Os valores lógicos estão associados a valores numéricos.



Modificadores de tipo

Tipo-base	Modificador(es)
int	short, long, signed, unsigned
char	signed, unsigned
double	long

Modificadores de tipo

Tipo	Tamanho (Bytes)	Intervalo de valores
char ou signed char	1	-128 a 127
unsigned char	1	0 a 255
int ou signed int	4	-2147483648 a 2147483647
unsigned int	4	0 a 4294967295
short int ou signed short int	2	-32768 a 32767
unsigned short int	2	0 a 65535
long int ou signed long int	4	-2147483648 a 2147483647
unsigned long int	4	0 a 4294967295

Identificadores

- Liguagem C é uma linguagem *case sensitive*.
  - “**a**” e “**A**” representam identificadores diferentes.
  - “**nota**” e “**NOTA**” também são identificadores diferentes entre si.

Variáveis

- Variáveis podem ser modificadas ao longo da execução;
- Na linguagem C a declaração de uma variável segue a sintaxe:

```
tipo_de_dado nome_da_variável ;
```

Variáveis

Exemplo:

```
#include <stdio.h>
int main ( ) // função principal
{
    // declarações de variáveis
    int horas;
    float salario;
    double dias, desconto;
    char nome [50];
    char sexo;
    ...
}
```

Constantes

const

- sintaxe de declaração:

```
const <nome da constante> = <valor da constante>;
```

- Exemplo:

```
const NOTA_MAXIMA = 100;
```

Constantes

#define

- sintaxe de declaração:

```
#define <nome da constante> <valor da constante>
```

- Exemplos:

```
#define MAX 40

#define VALOR_INF 5.7
```

Novos Tipos

- Sintaxe:

```
typedef <tipo original> <nome alternativo do tipo>
```

- Exemplos:

```
typedef int inteiro; //inteiro torna-se nome alternativo de int

typedef char caracter;//caracter torna-se nome alternativo de char
```

- Conforme esses dois exemplos, as seguintes declarações passam a ser possíveis:

```
inteiro valor_um, valor_dois;

caractere simbolo;
```