

# DECLARAÇÃO DE VARIÁVEIS E TIPO DE DADOS

LICENCIATURA EM ENGENHARIA  
INFORMÁTICA

## PALAVRAS RESERVADAS EM C

auto	double	int	struct
beak	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	Static	while

## VARIÁVEIS

- ◉ Nome dado a determinada posição de memória que recebe um valor de determinado tipo.
- ◉ Identificadas por nomes
- ◉ Especificadas por tipo
  - Inteiros, reais, caracteres
- ◉ Constantes - Um determinado valor fixo não modificável durante a execução de um programa

## VARIÁVEIS

### ◉ Identificadas por nomes

- Devem começar com uma letra ou um underscore (\_)
- Os caracteres seguintes podem ser letras, numero ou (\_)
- Não podem ser iguais a palavras reservadas
- Não podem ser iguais ao nome da função ou bibliotecas

# VARIÁVEIS - CONSTANTES

- Um determinado valor fixo não modificável durante a execução de um programa

```
Idade = 18;  
Nota = 12;  
Pp1 = 8;  
Total = nota + pp1;
```

# Declaração de Variáveis

Numéricas

- Números inteiros e reais (casas decimais)
- Usadas em operações matemáticas

Caracteres

- Conjunto de caracteres

Alfanuméricas

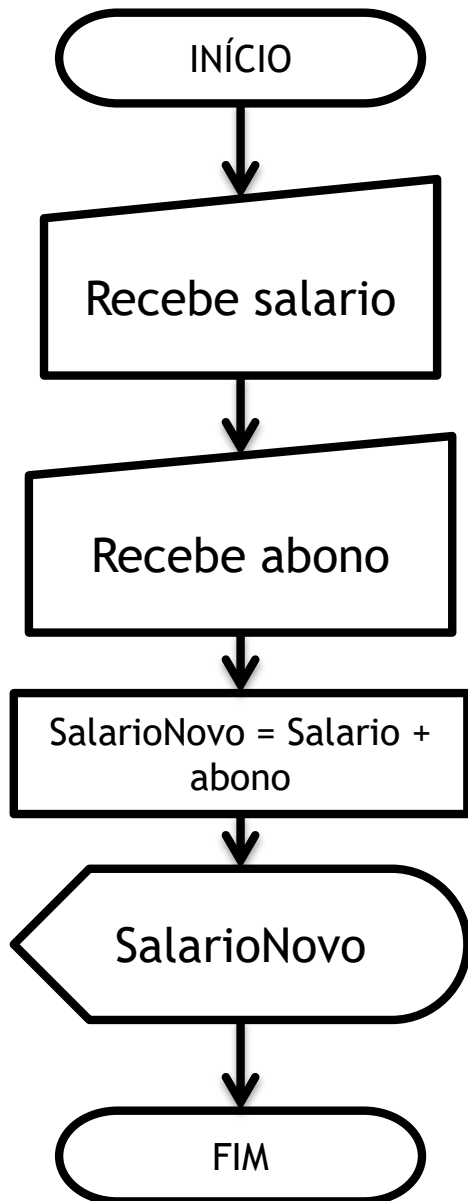
- Para números e/ou letras

Lógicas

- Verdadeiro ou falso

# EXERCÍCIOS

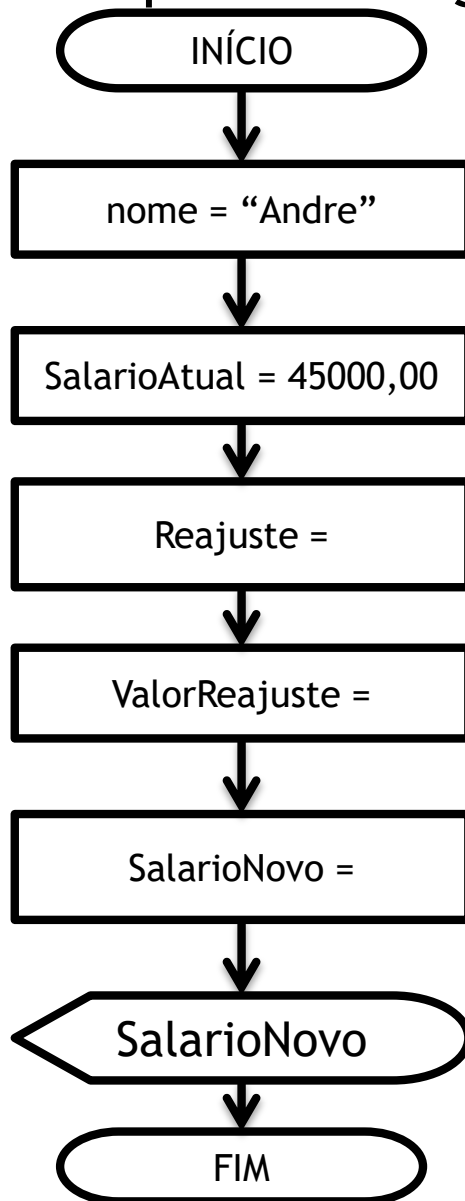
- ◉ O que é uma constante. Dê dois exemplos
- ◉ O que é uma variável. Dê dois exemplos
- ◉ Faça um teste de mesa no diagrama de código abaixo e preencha a tabela ao lado com os dados do teste



Salario	Abono	SalarioNovo
30000,00	2500,00	
60000,00		



- O André recebeu 5% de reajuste de salário. Complete o diagrama



# DECLARAÇÃO DE VARIÁVEIS

```
Tipo_da_variavel lista_de_variáveis;
```

## ◉ Tipo

- define o tamanho em bytes que ocupa na memória do computador e o intervalo de valores que pode armazenar

## ◉ Nome

- O primeiro caracter deverá ser uma letra ou *underline*(\_)

## ◉ Valor (opcional)

## VARIÁVEIS

- ◉ Funções distintas podem ter variáveis locais com o mesmo nome
- ◉ Se uma função declarar uma variável com o mesmo nome de uma variável global, está variável local ocultará a global, que não poderá ser acedida

# TIPOS DE DADOS

- Os tipos associados as variáveis indicam o nº de bytes que irão ser utilizados para guardar um valor na variável.

Tipo	descrição	Bytes	Intervalo de valores
char	Um único caracter	1	0 a 255
int	Números inteiros (sem casas decimais)	4	-2147483648 a +2147483647
float	Números em ponto flutuante com precisão simples (7 casas decimais)	4	$3,4 \cdot 10^{-38}$ a $3,4 \cdot 10^{38}$
double	Números em ponto flutuante com precisão dupla (15 casas decimais)	8	$1,7 \cdot 10^{-308}$ a $1,7 \cdot 10^{+308}$

Tipo	bits	scanf	intervalo	
char	8	%c	-128	127
unsigned char	8	%c	0	255
signed char	8	%c	-128	127
int	16	%i	-32.768	32.768
unsigned int	16	%u	0	65.535
signed int	16	%i	-32.768	32.768
short int	16	%hi	-32.768	32.768
unsigned short int	16	%hu	0	65.535
signed short int	16	%hi	-32.768	32.768
long int	32	%li	-2.147.483.648	2.147.483.647
signed long int	32	%li	-2.147.483.648	2.147.483.647
unsigned long int	32	%lu	0	4.294.967.295
float	32	%f	3.4E-38	3.4E+38
double	64	%lf	1,7E-308	1,7E+308
long double	80	%Lf	3,4E-4932	3,4E+4932

# DECLARAÇÃO DE VARIÁVEIS

Tipo\_da\_variavel lista\_de\_variáveis;

- ◉ Variáveis da lista de variáveis serão do mesmo tipo e separadas por virgula
- ◉ O tipo de dado *int* é *default em C*

```
#include<stdio.h>

int main(){

    int a, b, c;
    long d; //variavel do tipo long int
    float a;

    getchar();
    return(0);
}
```

## VARIÁVEIS

### ◉ Globais

- Definidas fora das funções
- Disponíveis para serem usadas em todo o programa
- Visíveis em todas as funções

### ◉ Locais

- Declaradas dentro de uma função
- Visíveis apenas na função em que foram criadas

### ◉ Lista de parâmetros

- Recebem valores externos
- Visíveis apenas na função em que foram criadas

```
#include<stdio.h>
/* prog2.c */
int a, b; /* 2 variáveis globais do tipo int */

int main(){

    int a, b, c; /* 3 variáveis locais do tipo int */
    double d; /* 1 variável local do tipo double */
    a = 5; /* atribuição de valor a variavel local */
    getchar();
    return(0);
}
```



```
#include<stdio.h>

int main(){

    int i; /* i é variável do tipo inteira */
    char ch1, novo_char; /* ch1, novo_char variáveis do tipo char */
    float pi, raio, perimetro;
    double total, k123;

    getchar();
    return(0);
}
```

# INICIALIZAÇÃO DE VARIÁVEIS

Tipo\_da\_variavel nome\_da\_variável = constante;

- ◉ Definição de um valor inicial para a variável;
- ◉ A inicialização pode ser com uma constante, expressão ou função
- ◉ Caso as variáveis globais não forem inicializadas recebem o valor zero (0)
- ◉ As locais devem ser inicializadas sob pena de resultarem em valores incorretos

```
int pp1=7, pp2=pp1 + 10;  
char letra = 'M';  
float media = (pp1 + pp2)/2;  
double notaFinal= 15.781;
```

# DIRECTIVA #DEFINE

- ◉ Utilizada para definição de macros
- ◉ As macros representam valores constantes
- ◉ Após a sua definição, toda a ocorrência do seu nome no programa é substituída pelo valor definido
- ◉ Por convenção os nomes de macros são escritos em letras maiúsculas

Sintaxe:

```
#define nome valor
```

Exemplo:

```
#define PI 3.14159
```

...

```
area = PI * (raio * raio);
```

Código	Significado
\b	Retrocesso (“back”)
\f	Alimentação de formulário (“form feed”)
\n	Nova linha (“new line”)
\t	Tabulação horizontal (“tab”)
\”	Aspas
\’	Apóstrofe
\0	Nulo (0 em decimal)
\\	Barra invertida
\v	Tabulação vertical
\a	Sinal sonoro (“beep”)
\N	Constante octal (N é o valor da constante)
\xN	Constante hexadecimal (N é o valor da constante)

# EXERCÍCIOS

- ◉ Escreva um programa que declare uma variável inteira global com valor 4. Declare localmente 4 variáveis inteiras no programa principal e atribua os valores 5, 10, 15 e 20 respectivamente. Declare 6 variáveis caracteres e atribua as letras i, m, e, t, r, o. O programa deverá retornar o seguinte:

“Sou estudante do imetro”  
Evolui as minhas notas em 5, 10, 15, 20

# EXERCÍCIOS

- Indique quais são as variáveis que têm nomes correctos e incorrectos:

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

```
int main(){
```

```
int idade; /*Correcto*/
```

```
int Num_Cliente; /*Correcto*/
```

```
float a1b2c3; /*Correcto*/
```

```
float 7a23c; /*Incorrecto: primeiro caracter é um dígito*/
```

```
char float; /*Incorrecto: float é uma palavra reservada*/
```

```
double vinte%; /*Incorrecto: % é um caracter não aceitável*/
```

```
char sim?não; /*Incorrecto: ? é um caracter não aceitável*/
```

```
int _alfa; /*Correcto mas não aconselhável*/
```

```
int _123; /*Correcto mas não aconselhável*/
```

```
int Num, NUM; /*Correcto por ser Case Sensitive*/
```

```
    getchar();
```

```
    return(0);
```

```
}
```

# ENTRADA E SAÍDA DE DADOS

LICENCIATURA EM ENGENHARIA  
INFORMÁTICA