



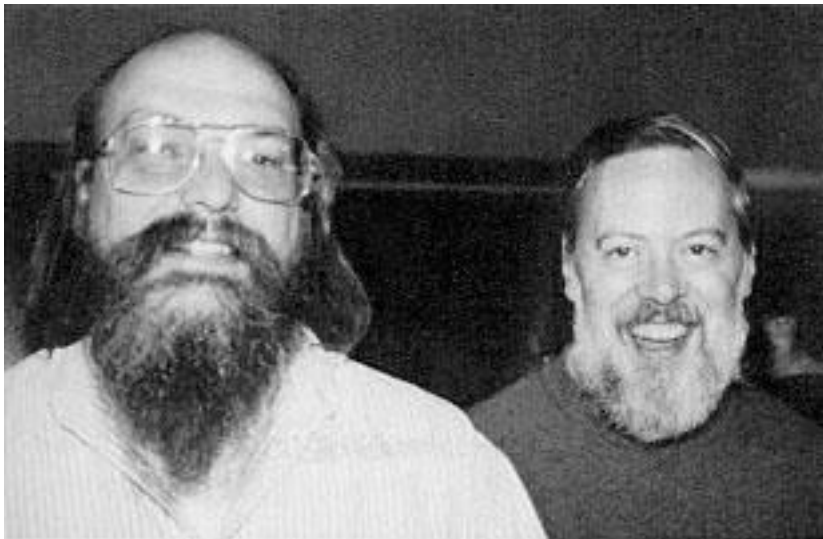
UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI
Campus Senador Helvídio Nunes de Barros - CSHNB
Curso Bacharelado em Sistemas de Informação

ALGORITMOS E PROGRAMAÇÃO I

Glauber Dias Gonçalves
ggoncalves@ufpi.edu.br

É A, é B, é C...

- C surgiu em 1972, e foi, durante muitos anos, a linguagem de programação mais popular.
 - Porque C tem esse nome?
 - Porque C foi e ainda é tão popular?



**Ken Thompson (esquerda)
Dennis Ritchie (direita)
desenvolvedores do Unix e
da linguagem C**

TIPOS BÁSICOS

- **char:** o valor armazenado é um caractere. Caracteres geralmente são armazenados em códigos (usualmente o código ASCII).
- **int:** número inteiro.
- **float:** número em ponto flutuante de precisão simples. São conhecidos normalmente como números reais.
- **double:** Número em ponto flutuante de precisão dupla.
- **void:** este tipo serve para indicar que um resultado não tem um tipo definido.

VARIÁVEIS

- Regras básicas para nomear variáveis:

- todo nome só pode conter letras e/ou dígitos;
- apenas o caractere símbolo "_" pode ser usado (acentuações são consideradas símbolos);
- todo primeiro caractere deve ser sempre uma letra;
- letras maiúsculas e minúsculas são consideradas caracteres diferentes;

- Declaração de variáveis:

- int i, idade, numero;
- float salario, altura;
- char sexo, letra;

- Atribuição:

- idade = 31;
- sexo = 'm';

VARIÁVEIS

- Regras básicas para nomear variáveis:

- todo nome só pode conter letras e/ou dígitos;
- apenas o caractere símbolo "_" pode ser usado (acentuações são consideradas símbolos);
- todo primeiro caractere deve ser sempre uma letra;
- letras maiúsculas e minúsculas são consideradas caracteres diferentes;

- Declaração de variáveis:

- int i, idade, numero;
- float salario, altura;
- char sexo, letra;

- Atribuição:

- idade = 31;
- sexo = 'm';

Caracteres usam
aspas simples.

VARIÁVEIS

- Regras básicas para nomear variáveis:

- todo nome só pode conter letras e/ou dígitos;
- apenas o caractere símbolo "_" pode ser usado (acentuações são consideradas símbolos);
- todo primeiro caractere deve ser sempre uma letra;
- letras maiúsculas e minúsculas são consideradas caracteres diferentes;

- Declaração de variáveis:

- int i, idade, numero;
- float salario, altura;
- char sexo, letra;

- Atribuição:

- idade = 31;
- sexo = 'm';

Caracteres usam
aspas simples.

Obs.: não só as variáveis mas toda a linguagem C é “**Case Sensitive**”, isto é, maiúsculas e minúsculas fazem diferença. Por exemplo: **Idade** ≠ **idade**, ou seja, são duas variáveis diferentes.

OPERADORES ARITMÉTICOS BÁSICOS

Operador	Símbolo	Exemplo
Adição	+	$a + b$
Subtração	-	$a - b$
Multiplicação	*	$a * b$
Divisão	/	a / b
Resto de Divisão Inteira	%	$a \% b$

OPERADORES RELACIONAIS E LÓGICOS

Operador	Símbolo	Exemplo
Igual	==	a == b
Diferente	!=	a != b
Maior	>	a > b
Maior ou igual	>=	a >= b
Menor	<	a < b
Menor ou igual	<=	a <= b
Conjunção	&&	a && b
Disjunção		a b
Negação	!	! c

BIBLIOTECAS

- Conjunto de funções prontas para uso
- Comando para incluir uma biblioteca no código:
 - `#include <nome da biblioteca>`
 - inserido no início do código
- Exemplo:
 - `stdio.h`: funções `scanf` e `printf` para E/S

COMANDO DE ENTRADA *SCANF*

- scanf (<info. de controle>, &<lista de variáveis>);
- Exemplos:
 - scanf ("%f", &salario);
 - scanf ("%d", &idade);
 - scanf ("%c", &letra);
 - scanf ("%s", nome);
 - scanf ("%d %f", &idade, &salario);
- O caractere **&** indica que o valor será armazenado no endereço de memória da variável.

COMANDO DE SAÍDA *PRINTF*

- printf (<info. de controle>, <lista de variáveis>);
- Exemplos:
 - printf ("Teste %% %")
 - "Teste % %"
 - printf ("%f", 40.345)
 - "40.345"
 - printf ("Um caractere %c e um inteiro %d", 'D', 120)
 - "Um caractere D e um inteiro 120"
 - printf ("%s eh um exemplo", "Este")
 - "Este eh um exemplo"
 - printf ("%s%d%%", "Juros de ", 10)
 - "Juros de 10%"

EXEMPLO (ESTRUTURA BÁSICA)

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

```
int main () {  
    printf ("Alô mundo!\n");  
    return 0;  
}
```

EXEMPLO (ESTRUTURA BÁSICA)

```
#include <stdio.h>

int main () {
    printf ("Alô mundo!\n");
    return 0;
}
```

Como executar esse código nos sistemas Windows ou Linux?

EXEMPLO: formatação saída (*printf*)

```
#include <stdio.h>

int main(){
    printf ("Teste %% %%\n");
    printf ("%f\n",40.345);
    printf ("Um caractere %c e um inteiro %d\n",'D',120);
    printf ("%s eh um exemplo\n","Este");
    printf ("%s%d%%\n","Juros de ",10);

    return 0;
}
```

ATIVIDADES EM SALA DE AULA

- Ler um número inteiro, exibir seus sucessor e antecessor.
- Ler uma quantidade de horas, exibir os valores equivalentes em minutos, segundos
- Ler o saldo de uma aplicação, exibir o novo saldo após um reajuste de 10%

ESTRUTURA DE SELEÇÃO

- Comando se

```
if (<condição>)  
    <comandos>;  
[ else  
    <comandos>; ]
```

Se **if** e **else** tiverem mais de um comando é necessário a utilização dos caracteres de chaves para marcar o início e fim do comando: { (início) e } (fim).

EXEMPLO

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

int main(){
    int idade;
    printf ("Digite um número:");
    scanf ("%d", &idade);
    if (idade >= 18)
        printf ("de maior\n");
    else
        printf ("de menor\n");

    return 0;
}
```

Estruturas Condicionais

1. Faça um algoritmo que, dadas as idades de duas pessoas, informe qual a pessoa mais velha ou se as duas tem a mesma idade.



Estruturas Condicionais

2. Crie um algoritmo que realize as seguintes atividades:

- a.** Solicite ao usuário dois valores inteiros.
- b.** Multiplique o menor valor lido pelo maior e some o resultado com o menor valor.
- c.** Imprima na tela o resultado.

DÚVIDAS?

