ALGORITMOS E ESTRUTURA DE DADOS

Curso de Engenharia de Software
Lucas Sampaio Leite

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Em C, toda variável precisa ser declarada antes de ser usada.
- A declaração informa ao compilador:
  - Tipo de dado (quanto espaço e como interpretar)
  - Nome da variável
  - Opcionalmente, um valor inicial

- Sintaxe sem inicialização: `tipo nome;`

- Sintaxe com inicialização: `tipo nome = valor;`

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Tipos primitivos mais comuns:
  - int → inteiros
  - float → ponto flutuante (precisão simples)
  - double → ponto flutuante (maior precisão)
  - char → caractere
  - void → ausência de tipo (usado em funções)

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Regras de nomeação de variáveis:
  - Nome deve começar com letra ou _
  - Não pode usar palavras reservadas (int, return, etc.)
- C é case-sensitive (idade ≠ Idade)
- Variáveis não inicializadas têm lixo de memória (valor indefinido)

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Declaração simples:

```
int idade;
float altura;
char letra;
```

- Declaração com inicialização:

```
int idade = 25;
float altura = 1.75;
char letra = 'A';
```

- Múltiplas variáveis na mesma linha:

```
int a = 10, b = 20, c;
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Em C, a entrada e saída padrão são feitas pela biblioteca:

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

- Saída (imprime dados na tela) → printf
  - printf("texto", variaveis);
  - %d → inteiro
  - %f → float/double
  - %c → caractere
  - %s → string

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

```c
#include <stdio.h>

int main(){

    printf("Hello, World!\n");

    int idade = 20;
    printf("Idade: %d\n", idade);

    return 0;
}
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Em C, a entrada e saída padrão são feitas pela biblioteca:

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

- Entrada (lê dados do teclado) → scanf
  - scanf("formato", &variavel);
  - Precisa usar & (endereço de memória), exceto para strings

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Erro comum:
```c
scanf("%d", idade);
```
- Correto:
```c
scanf("%d", &idade);
```
- Para String (sem &):
```c
char nome[50];
scanf("%s", nome);
```

Exercícios rápidos

INSTITUTO FEDERAL
Pernambuco

1. Leia um número inteiro e imprima:
   a) O número
   b) O dobro
   c) O triplo
2. Leia uma string e imprima a string digitada

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Operadores são símbolos que realizam operações sobre variáveis e valores.
- Principais categorias:
  - Aritméticos
  - Relacionais
  - Lógicos
  - Atribuição
  - Incremento/Decremento

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Operadores Aritméticos (utilizados para cálculos matemáticos):

Operador	Descrição	Exemplo
+	soma	a + b
-	subtração	a - b
*	multiplicação	a * b
/	divisão	a / b
%	módulo (resto)	a % b

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Operadores Aritméticos (utilizados para cálculos matemáticos):

Operador	Descrição	Exemplo
+	soma	a + b
-	subtração	a - b
*	multiplicação	a * b
/	divisão	a / b
%	módulo (resto)	a % b

Atenção!!! Divisão entre inteiros = resultado inteiro

Exercícios rápidos

INSTITUTO FEDERAL
Pernambuco

1. Leia dois números inteiros e mostre o resultado das operações de:
    a) Soma
    b) Subtração
    c) Multiplicação
    d) Divisão inteira

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Operadores Relacionais são usados para comparação (retornam 0 ou 1):

Operador	Significado
==	igual
!=	diferente
>	maior
<	menor
>=	maior igual
<=	menor igual

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Operadores Relacionais são usados para comparação (retornam 0 ou 1):

Operador	Significado
==	igual
!=	diferente
>	maior
<	menor
>=	maior igual
<=	menor igual

C não tem booleano nativo!!!
Em C, 0 representa falso e qualquer valor diferente de 0 representa verdadeiro.

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Operadores Lógicos são utilizados para combinar condições:

Operador	Significado
&&	AND (e)
||	OR (ou)
!	NOT (negação)

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Operadores de atribuição:

Operador	Equivalente
=	a = b
+=	a = a + b
-=	a = a - b
*=	a = a * b
/=	a = a / b

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Incremento e decremento:

Operador	Descrição
++	incrementa
--	decrementa

- Diferença importante:

```c
int x = 5;

printf("%d\n", x++); // usa 5, depois incrementa
printf("%d\n", ++x); // incrementa, depois usa
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Ordem de precedência de operadores (da maior para menor prioridade):
  1. () → parênteses
  2. ++, -- → incremento/decremento
  3. *, / → multiplicação/divisão
  4. +, - → soma/subtração
  5. <, <=, >, >= → relacionais
  6. ==, != → igual/diferente
  7. && → AND
  8. || → OR
  9. =,+=, -=, *=, /= → atribuição

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Quando operadores têm mesma precedência, entra a associatividade:
  - Esquerda → direita (mais comum)
  - Direita → esquerda (atribuição)

- Exemplos:

```c
int x = 10 - 5 / 2;
printf("%d", x);
```

```c
int x = (10 + 2) * 5;
printf("%d\n", x);
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Quando operadores têm mesma precedência, entra a associatividade:
  - Esquerda → direita (mais comum)
  - Direita → esquerda (atribuição)

- Exemplos:

```c
int x = 5 + 3 * 2 > 10;
printf("%d\n", x);
```

```c
int x = 10;
int y = 5;

int z = x > y && y < 3;
printf("%d\n", z);
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Quando operadores têm mesma precedência, entra a associatividade:
  - Esquerda → direita (mais comum)
  - Direita → esquerda (atribuição)

- Exemplos:

```c
int x = 5;

int y = x++ * 2;
printf("%d\n", y);
printf("%d\n", x);
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Estruturas condicionais permitem tomar decisões no código com base em condições (expressões que resultam em 0 ou 1).
- Principais estruturas:
  - if
  - if-else
  - else if
  - switch

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- if executa um bloco se a condição for verdadeira (≠ 0).

- Sintaxe:

```
if (condicao) {
    // código
}
```

- Exemplo:

```
int idade = 20;

if (idade >= 18) {
    printf("Maior de idade\n");
}
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- if-else permite um caminho alternativo.

- Sintaxe:

```
if (condicao) {
    // verdadeiro
} else {
    // falso
}
```

- Exemplo:

```c
int numero = 5;

if (numero % 2 == 0) {
    printf("Par\n");
} else {
    printf("Impar\n");
}
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- else if possibilita múltiplas condições.
- Sintaxe:

```
if (cond1) {
    // caso 1
} else if (cond2) {
    // caso 2
} else {
    // padrão
}
```

- Exemplo:

```c
int nota = 75;

if (nota >= 90) {
    printf("A\n");
} else if (nota >= 70) {
    printf("B\n");
} else if (nota >= 50) {
    printf("C\n");
} else {
    printf("Reprovado\n");
}
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- switch é usado quando há várias comparações com um mesmo valor.

- Sintaxe:

```
switch (variavel) {
    case valor1:
        // código
        break;
    case valor2:
        // código
        break;
    default:
        // padrão
}
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- switch é usado quando há várias comparações com um mesmo valor.

- Sintaxe:

```
switch (variavel) {
    case valor1:
        // código
        break;
    case valor2:
        // código
        break;
    default:
        // padrão
}
```

- Exemplo:

```c
int opcao = 2;

switch (opcao) {
    case 1:
        printf("Opcao 1\n");
        break;
    case 2:
        printf("Opcao 2\n");
        break;
    default:
        printf("Opcao invalida\n");
}
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- O operador ternário é uma forma compacta de escrever um if-else.
- Sintaxe: `condicao ? valor_se_verdadeiro : valor_se_falso;`
- Ele é uma expressão, ou seja, retorna um valor.
- Exemplo:

```
int idade = 20;

int status = (idade >= 18) ? 1 : 0;
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- O operador ternário é uma forma compacta de escrever um if-else.
- Sintaxe: `condicao ? valor_se_verdadeiro : valor_se_falso;`
- Ele é uma expressão, ou seja, retorna um valor.
- Exemplo:

```
int idade = 20;

int status = (idade >= 18) ? 1 : 0;
```

Como seria a estrutura if-else equivalente?

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

```c
int idade = 20;

int status = (idade >= 18) ? 1 : 0;
```

=

```c
int idade = 20;
int status;

if (idade >= 18) {
    status = 1;
} else {
    status = 0;
}
```

Exercícios rápidos

INSTITUTO FEDERAL
Pernambuco

1. Crie um programa que leia dois números e informe qual é o maior.
   a) com if-else
   b) com operador ternário
2. Crie um programa que leia a idade e classifique:
   a) Menor de idade (<18)
   b) Adulto (18–59)
   c) Idoso (60+)

Exercícios rápidos

INSTITUTO FEDERAL
Pernambuco

3. Crie um programa que leia uma nota (0–100) e imprima:
   a) A (>=90)
   b) B (>=70)
   c) C (>=50)
   d) D (<50)

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- Laços (loops) permitem executar um bloco de código várias vezes, enquanto uma condição for verdadeira.
- Estruturas em C:
  - for → quando sabemos o número de repetições
  - while → quando depende de uma condição
  - do-while → executa pelo menos uma vez

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- for é usado quando o número de iterações é conhecido.

- Sintaxe:

```
for (inicializacao; condicao; incremento) {
    // código
}
```

- Exemplo:

```
for (int i = 0; i < 5; i++) {
    printf("%d\n", i);
}
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- for é usado quando o número de iterações é conhecido.

- Exemplo:

```c
for (int i = 0; i < 5; i++) {
    printf("%d\n", i);
}
```

- Funcionamento:
  - inicialização → executa 1 vez
  - condição → verificada a cada repetição
  - incremento → executado ao final de cada loop

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- while executa enquanto uma condição for verdadeira.

- Sintaxe:

```
while (condicao) {
    // código
}
```

- Exemplo:

```
int i = 0;

while (i < 5) {
    printf("%d\n", i);
    i++;
}
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- do-while executa pelo menos uma vez, mesmo que a condição seja falsa.

- Sintaxe:

```
do {
    // código
} while (condicao);
```

- Exemplo:

```
do {
    printf("%d\n", i);
    i++;
} while (i < 5);
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- break e continue em C são comandos que alteram o fluxo de execução dentro de laços (for, while, do-while) e também no switch.
  - break → interrompe completamente o laço
  - continue → pula para a próxima iteração

- Exemplo:

```c
for (int i = 0; i < 10; i++) {
    if (i == 5) {
        break;
    }
    printf("%d\n", i);
}
```

Revisão da linguagem C

INSTITUTO FEDERAL
Pernambuco

- break e continue em C são comandos que alteram o fluxo de execução dentro de laços (for, while, do-while) e também no switch.
  - break → interrompe completamente o laço
  - continue → pula para a próxima iteração

- Exemplo:

```c
for (int i = 0; i < 5; i++) {
    if (i == 2) {
        continue;
    }
    printf("%d\n", i);
}
```

Exercícios rápidos

INSTITUTO FEDERAL
Pernambuco

1. Crie um programa que imprima os números de 1 a 10 usando for, while e do-while.
2. Crie um programa que leia 5 números e mostre a soma total.
3. Crie um programa que leia um número e mostre sua tabuada de 1 a 10.
4. Crie um programa que leia um número e calcule o seu fatorial. Ex: 5! = 5 * 4 * 3 * 2 * 1 = 120

Dúvidas

INSTITUTO FEDERAL
Pernambuco

ALGORITMOS E ESTRUTURA DE DADOS

Curso de Engenharia de Software
Lucas Sampaio Leite