

Introdução à Programação

Linguagem C

Nelson Carvalho Sandes

Centro de Ciências Tecnológicas - CCT
Universidade Federal do Cariri

2020

Tópicos

- 1 Vantagens
- 2 Processo de compilação
- 3 Instrução de saída em C
- 4 Operações matemáticas em C
- 5 Variáveis
- 6 Instrução de entrada em C
- 7 Exercícios

Tópicos

- 1 Vantagens
- 2 Processo de compilação
- 3 Instrução de saída em C
- 4 Operações matemáticas em C
- 5 Variáveis
- 6 Instrução de entrada em C
- 7 Exercícios



Vantagens da linguagem C

- C usa aspectos que não são explícitos em algumas linguagens de programação, como alocação de memória e utilização de ponteiros.
- Por conta disso, o estudante se sentirá confortável no aprendizado de outras linguagens de programação.
- É uma das linguagens mais usadas no mundo, de acordo com o site www.tiobe.com.

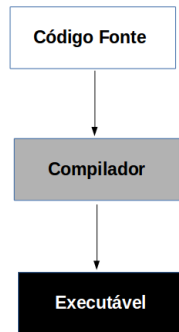


Tópicos

- 1 Vantagens
- 2 **Processo de compilação**
- 3 Instrução de saída em C
- 4 Operações matemáticas em C
- 5 Variáveis
- 6 Instrução de entrada em C
- 7 Exercícios

Compilação

- Em nossas aulas, criaremos algoritmos na linguagem de programação **C**, que é uma linguagem **compilada**.
- Em uma linguagem compilada, se utiliza um **compilador** para transformar o código fonte em um executável.
- Depois de gerar o executável, o usuário pode pedir para o sistema operacional executar o programa.



Tópicos

- 1 Vantagens
- 2 Processo de compilação
- 3 **Instrução de saída em C**
- 4 Operações matemáticas em C
- 5 Variáveis
- 6 Instrução de entrada em C
- 7 Exercícios

Instrução de saída em C

Saída de dados: instrução printf

- Abaixo, segue a estrutura de um programa em **C**.
- A instrução **printf** é um comando de saída de dados que imprime uma mensagem na tela.

```
#include <stdio.h>
int main() {
    printf("Hello world! \n");
    return 0;
}
```


Instrução de saída em C

Saída de dados: instrução printf

- Para imprimir informações com dados, é preciso utilizar símbolos especiais para indicar o tipo de dado que será representado.

Especificador	Representa
%c	Um único caractere
%d	Um número inteiro em base decimal
%ld	Um número inteiro longo em base decimal
%f, %lf	Um número real de precisão simples ou dupla
%s	Uma cadeia de caracteres

Instrução de saída em C

printf: Exemplo 1

■ Exemplo:

```
#include <stdio.h>

int main() {
    printf(" Ele tem 42 anos \n");
    printf(" Ele tem %d anos \n", 42);
    return 0;
}
```

Instrução de saída em C

printf: Exemplo 2

■ Exemplo:

```
#include <stdio.h>
int main() {
    printf(" O valor da taxa SELIC é de 0.065 \n");
    printf(" O valor da taxa %s é de %f \n", "SELIC", 0.065);
    return 0;
}
```

Tópicos

- 1 Vantagens
- 2 Processo de compilação
- 3 Instrução de saída em C
- 4 Operações matemáticas em C
- 5 Variáveis
- 6 Instrução de entrada em C
- 7 Exercícios

Operações matemáticas em C

- Em **C**, também é possível realizar operações matemáticas.
- Nesse contexto, usar os especificadores `%d` e `%f` pode ser muito útil.

```
#include <stdio.h>

int main() {
    printf("Soma: %d \n", 1 + 2 + 3);
    printf("Subtração: %d \n", 5 - 3);
    printf("Multiplicação: %f \n", 5 * 7.5);
    printf("Divisão: %d \n", 10/2);
    printf("Resto: %d \n", 10%2);
    return 0;
}
```

- Resultados: 6, 2, 37.50, 5 e 0 respectivamente.

Operações matemáticas em C

Operador de divisão

- Em **C**, a divisão entre dois números inteiros irá retornar o valor inteiro da divisão.
- Para se obter o valor da divisão, é preciso que pelo menos um dos números seja ponto flutuante.

```
#include <stdio.h>
int main() {
    printf("Divisão com inteiros: %d \n", 23/10);
    printf("Divisão: %f \n", 7.0/2);
    return 0;
}
```

- Resultados: 2 e 3.5 respectivamente.

Prioridades

- É importante notar que a divisão e a multiplicação tem prioridade sobre a adição e a subtração.

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

```
int main() {
    printf("%d \n", 2 + 6/3);
    printf("%d \n", 2 + 3*5 + 4);
    printf("%d \n", 4/2 + 3*2 + 1);
    return 0;
}
```

- Resultados: 4, 21, e 9 respectivamente.

Operações matemáticas em C

Prioridades

- Para dar prioridade para alguma operação, faz-se o uso dos parênteses. Outro fator importante é que as operações no computador são feitas da esquerda para direita.

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

```
int main() {
    printf("%d \n", (3 + 6)/3);
    printf("%d \n", (2 + 3)*(5 + 4));
    printf("%d \n", 16/(4 - 3)*(2 + 2));
    return 0;
}
```

- Resultados: 3, 45, e 64 respectivamente.

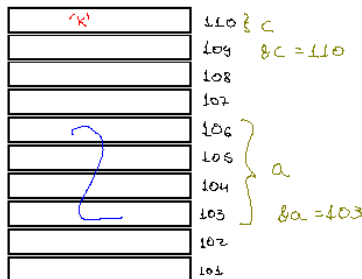
Tópicos

- 1 Vantagens
- 2 Processo de compilação
- 3 Instrução de saída em C
- 4 Operações matemáticas em C
- 5 Variáveis
- 6 Instrução de entrada em C
- 7 Exercícios

Variáveis

Armazenamento em memória

- Em computação, o conceito de variável caracteriza-se por ser uma região da memória que possui um **endereço** e tem a capacidade de **armazenar um valor**, seja este numérico, alfabético, alfanumérico ou lógico. Além disso, podemos acessar a região de memória através de um **identificador**.



Variáveis

Exemplo de criação de variável inteira

- Abaixo, segue um exemplo de como criar uma variável inteira em **C**.
- O sinal de **=** é um operador de **atribuição**.

```
#include <stdio.h>
int main() {
    int idade;
    idade = 32;
    printf("O valor da variável é: %d \n", idade);
    return 0;
}
```

Identificador

- O "nome" de uma variável é chamado de identificador.
- Regras para nomear uma variável:
 - 1 Iniciar com letra
 - 2 Não possuir caracteres especiais (#, %)
 - 3 Não possuir acentos (á, ú, â, ã)
 - 4 Não conter espaços (_ pode ser usado)
 - 5 Não pode ser uma palavra reservada (printf, scanf)
- Apesar de não ser regra, é uma convenção não iniciar variáveis com letras maiúsculas.

Variáveis

Declarações de diferentes tipos de variáveis

■ Exemplo de declarações de variáveis C.

```
#include <stdio.h>
int main() {
    int idade = 32;
    float altura = 1.69;
    char letra = 'c';
    printf("valores: %d, %f, %c \n", idade, altura, letra);
    return 0;
}
```

Variáveis

Operações matemáticas com variáveis

- Assim como números, pode-se fazer operações matemáticas com variáveis.

```
#include <stdio.h>
int main() {
    int a, b, c, soma;
    a = 1;
    b = 5;
    c = 10;
    soma = a + b + c;
    printf("Soma de a, b, c: %d \n", soma);
    return 0;
}
```

Variáveis

Operações matemáticas com variáveis

- Outros exemplos:

```
#include <stdio.h>

int main() {
    int a, b;
    a = 3;
    b = 5;
    printf("Soma: %d \n", a + b);
    printf("Subtração: %d \n", a - b);
    printf("Multiplicação: %d \n", a * b);
    printf("Divisão: %d \n", a/b);
    return 0;
}
```

Operações matemáticas com variáveis

Sequência de tempo

```
#include <stdio.h>
int main() {
    int divida, compra;
    divida = 0;
    compra = 100;
    divida = divida + compra;
    compra = 200;
    divida = divida + compra;
    compra = 300;
    divida = divida + compra;
    printf("Dívida: %d \n", divida);
    return 0;
}
```


Exercícios

- 1 Faça um algoritmo que exiba o resultado da operação $2a + 3b$. Adote $a = 3$ e $b = 5$
- 2 Faça um algoritmo que dê um aumento de 10% em um salário. Considere que o salário é de R\$1.500.

Tópicos

- 1 Vantagens
- 2 Processo de compilação
- 3 Instrução de saída em C
- 4 Operações matemáticas em C
- 5 Variáveis
- 6 Instrução de entrada em C
- 7 Exercícios



Instrução de entrada em C

Entrada de dados: instrução scanf

- A instrução **scanf** permite que um valor seja lido do teclado e armazenado em uma variável.
- Exemplo:

```
#include <stdio.h>
int main() {
    int a;
    printf("Digite um número inteiro. \n");
    scanf("%d", &a);
    printf("Número digitado: %d \n", a);
    return 0;
}
```

Instrução de entrada em C

Entrada de dados: instrução scanf

- Exemplo:

```
#include <stdio.h>
int main() {
    int numero;
    char letra;
    printf("Digite um número inteiro e uma letra. \n");
    scanf("%d %c", &numero, &letra);
    printf("Valores digitados: %d e %c \n", numero, letra);
    return 0;
}
```

Tópicos

- 1 Vantagens
- 2 Processo de compilação
- 3 Instrução de saída em C
- 4 Operações matemáticas em C
- 5 Variáveis
- 6 Instrução de entrada em C
- 7 Exercícios



Exercícios

- 1 Faça um programa que peça dois números inteiros. Imprima a soma desses dois números na tela.
- 2 Escreva um programa que leia um valor em metros e exiba em milímetros.
- 3 Escreva um programa que leia a quantidade de dias, horas, minutos e segundos do usuário. Calcule o total em segundos.
- 4 Faça um programa que calcule o aumento de um salário. Ele deve solicitar o valor do salário e a porcentagem do aumento. Exiba o valor do aumento e do novo salário.

Exercícios

- 1 Escreva um programa que solicite o preço de uma mercadoria e o percentual de desconto. Exiba o valor do desconto e o preço a pagar.
- 2 Escreva um programa que calcule o tempo de uma viagem de carro. Pergunte a distância a percorrer e a velocidade média esperada para a viagem.
- 3 Escreva um programa que converta uma temperatura digitada em Cº em Fº. A fórmula para conversão é: $F = \frac{9C}{5} + 32$