

Conteúdo

- Introdução a linguagem C
- Variáveis e seus tipos
- Declarações
- Atribuição
- Comandos de entrada e saída de dados
- Operadores aritméticos, relacionais e lógicos

Competências

- (1) Entender sobre a construção de programas para a solução de problemas computacionais, usando uma linguagem estruturada e o contexto de dados utilizando meios de interconexão com o usuário.

1. Introdução à linguagem C

Criada pelo cientista da computação Dennis Ritchie, em 1972. Embora tenha sido pensada com o propósito exclusivo de ser usada no desenvolvimento de uma nova versão do sistema operacional Unix, hoje é aplicada nos mais variados tipos de projeto.

Considerada uma linguagem de alto nível genérica, a linguagem C pode ser usada em diversos tipos de projeto, como a criação de aplicativos, sistemas operacionais, drivers, entre outros. Trata-se de uma linguagem estruturada que se tornou muito popular nos anos 80 - tanto que é difícil encontrar arquiteturas para as quais não existam compiladores para a linguagem C, o que garante o seu elevado nível de portabilidade.

Uma das grandes vantagens dessa linguagem é a capacidade de gerar códigos rápidos com um tempo de execução baixo. Além disso, a programação em C é bastante simplificada, pois sua estrutura é simples e flexível. Tendo isso em mente, podemos dizer que as principais características da linguagem C são a portabilidade, a geração de código eficiente, a simplicidade, a confiabilidade e facilidade de uso.

A linguagem C ainda é uma das mais populares do mercado devido às diversas vantagens que apresenta.

2. Compiladores e Dev-C++

Um compilador é um programa de sistema que traduz um programa descrito em uma linguagem de alto nível para um programa equivalente em código de máquina para um processador. Em geral, um compilador não produz diretamente o código de máquina, mas sim um programa em linguagem simbólica semanticamente equivalente ao programa em linguagem de alto nível. O programa em linguagem simbólica é então traduzido para o programa em linguagem de máquina através de montadores.

O Dev-C++ é um software que oferece um ambiente de desenvolvimento integrado para o desenvolvimento de aplicações. O programa possui todas as funcionalidades padrões necessárias para a escrita, compilação, debugging e execução de programas em C.

O Dev-C++ é gratuito e seu download pode ser feito em <https://sourceforge.net/projects/orwelldevcpp/>

3. Um programa em Portugol

Elaborar um algoritmo que leia dois números inteiros e calcule sua soma.

```
ALGORITMO Soma_de_dois_números
VAR
    INTEIRO: a, b, soma;
INICIO
    ESCREVA("Informe o valor de a: ");
    LEIA(a);

    ESCREVA("Informe o valor de b: ");
    LEIA(b);

    soma ← a + b;

    ESCREVA ("Resultado da soma: ", soma);
FIM
```

4. Um programa em C

```
#include <stdio.h>

int main()
{
    int a, b, soma;

    printf("Informe o valor de a: ");
    scanf("%d", &a);

    printf("Informe o valor de b: ");
    scanf ("%d", &b);

    soma = a + b;

    printf("Resultado da soma: %d", soma);

    return (0);
}
```

5. Variáveis e seus tipos

Em um programa, existe a necessidade de se guardar valores na memória, e isso é feito através de **variáveis**, que podem ser definidas simplificadaamente como nomes que se referem a lugares na memória onde são guardados valores. Ao declararmos uma variável, não apenas estamos reservando um espaço de memória, como também estamos associando um nome a ele, o **identificador**.

A linguagem C apresenta os seguintes tipos de dados fundamentais:

Tipo	Descrição
int	Para números inteiros entre -2147483648 e 2147483647. Utiliza 4 bytes.
char	Para caracteres individuais do padrão ASCII. Utiliza 1 byte.
float	Para números reais entre 10^{-38} e 10^{38} . Utiliza 6 bytes, precisão de 8 dígitos.
double	Para números reais entre 10^{-4932} e 10^{4932} . Utiliza 8 bytes, precisão de 15 dígitos.
bool	Para indicar true (verdadeiro) ou false (falso). Utiliza 2 bytes.

6. Declarações

Em C, para utilizar uma variável, ela deve ser primeiramente declarada, ou seja, devemos requisitar o espaço necessário para essa variável. Após reservar um espaço na memória, o computador irá associar a ele o nome da variável. Se você não declarar uma variável e tentar utilizá-la, o compilador irá avisá-lo disso e não continuará a compilação.

Em linguagem C a declaração de uma variável pode ser feita utilizando-se a seguinte sintaxe:

```
tipo_variavel nome_variavel
```

Exemplos:

```
int idade;  
float peso;  
char genero;  
char nome[30];  
double rendimento;
```

Quando é preciso definir mais de uma variável do mesmo tipo, pode-se fazer isso em uma mesma instrução, apenas separando o nome das variáveis por vírgula, como os exemplos a seguir:

```
int idade, matricula;  
float vCelcius, vKelvin;  
char genero, sexo;
```

7. Comandos de atribuição

Para armazenar dados em variáveis, é preciso fazer atribuição. Para isso, deve-se utilizar o operador de atribuição, que na linguagem C é o sinal de igualdade, "=". Veja a seguir alguns exemplos do uso de atribuição.

```
int idade, matricula;  
idade = 30;  
matricula = 123659;
```

A atribuição também pode ser feita no momento da declaração da variável, como mostrado a seguir.

```
float preco = 42.9;
```

Outros operadores de atribuição podem ser utilizados, conforme mostrado na tabela a seguir.

Operador	Operação matemática
=	Atribuição simples
+=	Atribuição acumulando por soma
-=	Atribuição acumulando por subtração
*=	Atribuição acumulando por multiplicação
/=	Atribuição acumulando por divisão
%=	Atribuição acumulando por módulo

8. Comandos de entrada e saída

8.1 Função printf()

A função printf() permite realizar a impressão de textos no monitor, ou seja, é responsável pela saída de informações. Exemplo:

```
printf("Este é um programa em C");
```

A função printf também pode imprimir o conteúdo de variáveis. Para isso, você deve especificar onde o valor da variável deve ser impresso através de uma especificação de formato.

Exemplo 1:

```
int teste;  
teste = 42;  
printf ("A variável 'teste' contém o número %d.", teste);
```

Resultado: "A variável 'teste' contém o número 42."

Exemplo 2:

```
float pi;  
pi = 3.1415;  
printf ("O valor de pi é %f.", pi);
```

Resultado: " O valor de pi é 3.1415."

Os formataadores de tipo podem ser vistos na tabela a seguir.

Formato	Tipo de variável
%d	int
%s	char
%f	float, double

Também é possível formatar a saída de uma variável float ou double.

Exemplo 1:

```
float total = 300 + 400;  
printf("Total da conta: %3.2f", total);
```

Resultado: "Total da conta: 700.00"

8.2 Função scanf()

A função scanf() lê dados da entrada padrão (teclado) e os guarda em variáveis do programa. Assim como para printf(), usamos uma string de formatação para especificar como serão lidos os dados. Exemplo:

Exemplo 1:

```
int a;  
scanf ("%d", &a);  
  
float preco;  
scan ("%f", &preco);
```

9. Operadores aritméticos, relacionais e lógicos

9.1 Operadores aritméticos

Operador	Operação matemática
+	Adição
-	Subtração.
*	Multiplicação
/	Divisão
%	Módulo (obtem o resto da divisão)
++	Incremento unário
--	Decremento unário

9.2 Operadores relacionais

Operador	Nome	Exemplo
==	Igualdade	a == b
>	Maior que	a > b
>=	Maior ou igual que	a >= b
<	Menor que	a < b
<=	Menor ou igual que	a <= b
!=	Diferente de	a != b

9.3 Operadores lógicos

Operador	Representação em C	Exemplo
E (conjunção)	&&	x > 1 && x < 19
OU (disjunção)	(duas barras verticais)	x == 1 x == 2
NÃO (negação)	! (exclamação)	! Continuar