
Lógica de Programação

Linguagem C

Objetivos

- ❑ Conhecer a linguagem C para criação de algoritmos;
- ❑ Reconhecer os tipos de dados e declarar os identificadores;
- ❑ Utilizar comandos de entrada e saída.

Turbo C++

- ❑ O ambiente Turbo C++ integra os recursos de um compilador com um editor de textos, que será selecionado através de opções no menu.
- ❑ O Turbo C++ possui um menu de comandos com algumas opções que possibilitam executar diversas tarefas operacionais.

Etapas para a programação no C++

As três etapas de desenvolvimento da linguagem C:

- ❑ Criação do programa fonte (digitação do texto)
- ❑ Salvar o arquivo
- ❑ Compilação desse programa, para a sua tradução para código executável (Compile)
- ❑ Execução do código produzido (Run)

Estrutura da programação em Turbo C++

```
/* Declaração de bibliotecas */  
int main(int argc, char *argv[])  
{  
    corpo do programa;  
}
```

Bibliotecas

- `#include` - insere o conteúdo de um arquivo de texto no arquivo corrente. Esses arquivos são usualmente designados por cabeçalhos e têm a extensão `.h`.

Alguns exemplos:

- `#include <math.h>` - Insere o conteúdo do arquivo `math.h` com a declaração das funções matemáticas da biblioteca standard.
- `#include <stdio.h>` - Idem para as funções standard de entrada/saída.
- `#include <iostream.h>` - ele contém declarações necessárias ao uso do objeto **`cout`** e do operador de inserção `<<`
- `#include <conio.h>` - contém o protótipo das funções `getch()` e `getche()`.

Código de Barra Invertida

Código	Significado
\b	Retrocesso
\f	Alimentação de formulário
\n	Nova linha
\r	Retorno de carro
\t	Tabulação horizontal
\"	Aspas
\'	Apóstrofo
\0	Nulo
\\	Barra invertida
\v	Tabulação vertical
\a	Sinal sonoro
\N	Constante octal (onde N é uma constante octal)
\xN	Constante hexadecimal (onde N é uma constante hexadecimal)

Tipo de dados

Tipo de dados	Em Português Estruturado	Tamanho (bytes)	Limite inferior	Limite superior
char	Caractere	1	-128	127
short int	Inteiro	2	-32768	32767
unsigned short int	Inteiro	2	0	65535
int	Inteiro	4	-2^{31}	$+2^{31} - 1$
long int	Inteiro	4	-2^{31}	$+2^{31} - 1$
float	Real	4	$-3.2 \times 10^{\pm 38}$	$+3.2 \times 10^{\pm 38}$
double	Real	8	$-1.7 \times 10^{\pm 308}$	$+1.7 \times 10^{\pm 308}$

Operadores de atribuição

Operador	Descrição
=	Atribuição
+=	Atribuição e soma
-=	Atribuição e subtração
*=	Atribuição e multiplicação
\=	Atribuição e divisão
%=	Atribuição de Módulo

Operadores Aritméticos

Operador	Descrição	Exemplo
+	Adição	$x = y + z;$
-	Subtração	$x = y - z;$
*	Multiplicação	$x = y * z;$
/	Divisão	$x = y / z;$
%	Módulo	$x = y \% z;$

Linguagem C

Exponenciação

`pow (base, expoente)`

Ex. $3^2 = \text{pow}(3,2)$

Radiciação

`sqrt(variavel)`

Ex. `sqrt (b)`

`sqrt (25)`

Linguagem C

Operadores Relacionais

> maior que
>= maior ou igual
< menor
<= menor ou igual
== igual
!= não igual

Operadores Lógicos

&& and
|| ou
! not

Marcador de Posição com formato

Formato	Tipo de dado
%d	Decimal inteiro
%f	Ponto flutuante
%c	Apenas um caractere
%s	Cadeia de caracteres

Linguagem C

❑ Comando de entrada

`scanf (<formato>, &<variavel>)`

Obs : & é o operador de endereçamento

Ex: `scanf ("%c %d %f", &x,&y,&z);`

Este comando lê 3 dados digitados pelo usuário, e os armazena nesta mesma ordem, nos endereços de memória em que foram criadas as variáveis x,y,z.

Linguagem C

- ❑ Comando de saída
- ❑ `printf (<formato>, <variavel>)`

Ex: `printf(“\n O valor total é : %f”, res);`

- ❑ Exemplo de declaração de variáveis

`char X;`

`int Y;`

`float Z;`

Exemplo 1

Algoritmo: Entre com sua idade e depois mostrar a idade na tela.

```
#include <studio.h>
#include <conio.h>
int main(int argc, char *argv[])
{
    int idade ; /* declaração de variável */
    clrscr( ); /* função predefinida para limpar a tela */
    printf (" Digite sua idade: "); /* Apresenta a mensagem entre aspas na tela */
    scanf (" %d",&idade ); /* lê via teclado um valor que é colocado em idade */
    printf ( "Sua idade é: %d",idade );
    system("PAUSE"); /* função predefinida, que congela a tela */
    return 0; /* função predefinida, espera uma tecla ser pressionada */
}
```


Exemplo 2

Algoritmo: Soma de dois números.

```
#include <stdio.h>
#include <conio.h>
int main(int argc, char *argv[])
{
    float soma;
    float num1,num2; /* declaração de variável */
    clrscr( ); /* função predefinida para limpar a tela */
    printf("\n Digite o 1o. numero : "); /*Apresenta a mensagem entre aspas na tela */
    scanf("%f", &num1); /*Lê os valores digitados no teclado*/
    printf("\n Digite o 2o. numero : "); /*Apresenta a mensagem entre aspas na tela */
    scanf("%f",&num2); /*Lê os valores digitados no teclado*/
    soma = num1 + num2;
    printf("\nA soma e : %f \n", soma); /*Apresenta a mensagem entre aspas na tela */
    system("PAUSE"); /* função predefinida, que congela a tela */
    return 0; /* função predefinida, espera uma tecla ser pressionada */
}
```

Exercícios

1) Construa um algoritmo que calcule a média aritmética entre 4 notas oferecidas pelo usuário.

2) Crie um algoritmo que calcule a área de um triângulo.

Exercícios

3) Elaborar um algoritmo que efetue o calculo da quantidade de litros de combustível gastos em uma viagem, utilizando-se um automóvel que faz 12km por litro. Para obter o calculo, o usuário deverá fornecer o Tempo Gasto na viagem e a Velocidade Média durante a mesma. Desta forma, será possível obter a distancia percorrida com a fórmula:

Distância = tempo x velocidade

Tendo o valor da distancia, basta calcular a quantidade de litros de combustível utilizada na viagem com a fórmula:

Litros usados = Distância / 12

O programa deverá apresentar os valores da Velocidade Média, Tempo Gasto na viagem, a Distância percorrida e a Quantidade de litros utilizada na viagem.

Bom trabalho!