

Linguagem de Programação

(ILP-010)

Prof. Dr. Silvio do Lago Pereira

Departamento de Tecnologia da Informação

Faculdade de Tecnologia de São Paulo



Contato

- **Sala:** 623 – Bloco A
- **E-mail:** slagop@gmail.com
- **Página:** www.ime.usp.br/~slago
 - Tópicos
 - Compilador
 - Bibliografia
 - Avaliação
 - Notas
 - Cronograma



Curso

- **Objetivo:** Implementar algoritmos em linguagem C.
- **Tópicos:** E/S básica, seleção, repetição, funções, vetores, estruturas, ponteiros e arquivos.
- **Avaliação:**
 - **Trabalhos** em dupla, com consulta: T1 e T2
 - **Provas** individuais, sem consulta: P1 e P2
 - **Média** = $0,4 * (P1 + P2) + 0,1 * (T1 + T2)$
 - **Observações:**
 - Aprovação requer média no mínimo **6,0** (além de frequência).
 - Quem não atingir a média poderá fazer **uma** prova substitutiva, cuja nota substituirá a menor nota de prova (apenas uma delas).

Introdução

(ILP-010)





Introdução

Linguagem C

é considerada uma das linguagens de programação mais eficientes que existem.

Histórico:

- Criada em 1972, por Dennis Ritchie, nos Laboratórios Bell.
- Evolução da linguagem BCPL ($\rightarrow B \rightarrow \mathbf{C} \rightarrow C++, \text{JAVA}, C\#, \dots$).

Algumas características:

- **Flexível**: usada na criação de diversos tipos de aplicações.
- **Portátil**: pode ser executada em diferentes plataformas.
- **Eficiente**: proporciona velocidade de execução e economia de memória.

Algumas aplicações:

- Jogos.
- Controladores.
- Compiladores.
- Sistemas operacionais.





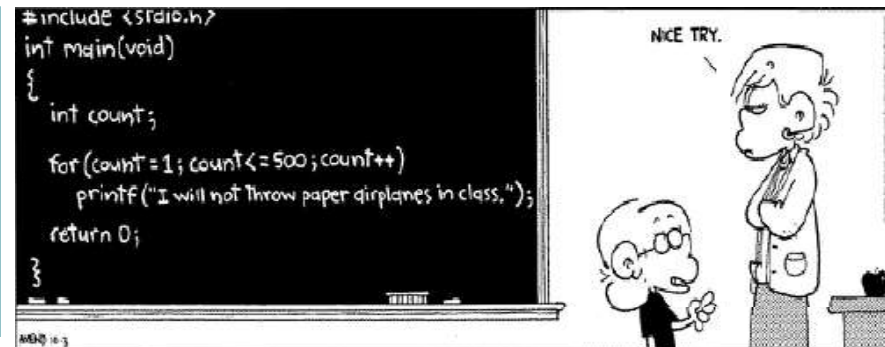
Introdução

Programa em C

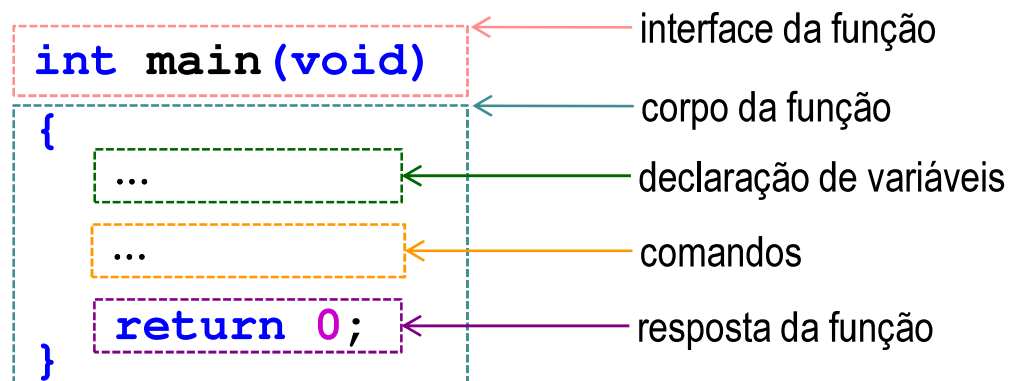
é um conjunto de uma ou mais funções, em que uma delas é chamada **main**.

Função:

- Bloco de código independente e reusável.
- A função principal em C é chamada **main**.
- A execução sempre inicia na função principal.



Função principal:





Introdução

Exemplo 1. Índice de massa corpórea (IMC)

Uma pessoa está obesa se seu IMC (peso dividido pela altura ao quadrado) é superior a 30. Dados o peso e a altura de uma pessoa, informe se ela está obesa.

```
/* OBESO.C - determina se uma pessoa está obesa */
#include <stdio.h> // entrada e saída padrão
#include <math.h>  // funções matemáticas

int main(void) {
    float peso, altura, imc;
    printf("Qual o peso e a altura? ");
    scanf("%f %f", &peso, &altura);
    imc = peso/pow(altura,2);
    printf("IMC = %.1f\n", imc);
    if( imc<=30 ) printf("Nao esta obesa!\n");
    else printf("Esta obesa!\n");
    return 0;
}
```



Introdução

Compilador *Pelles C*

é o software que usaremos para criar e executar programas codificados em C.

```
/* OBESO.C - determina se uma pessoa está obesa */  
  
#include <stdio.h> // entrada e saída padrão  
#include <math.h> // funções matemáticas  
  
int main(void) {  
    float peso, altura, imc;  
    printf("Qual o peso e a altura? ");  
    scanf("%f %f", &peso, &altura);  
    imc = peso/pow(altura,2);  
    printf("IMC = %.1f\n",imc);  
    if( imc<=30 ) printf("Nao esta obesa!\n");  
    else printf("Esta obesa!\n");  
    return 0;  
}
```

The screenshot shows the Pelles C IDE interface. The main window displays the source code for a program named 'obeso.c'. The code calculates the BMI (IMC) and checks if a person is obese. The interface includes a menu bar (File, Edit, View, Project, Source, Tools, Window, Help), a toolbar, and a status bar at the bottom showing 'Project build started' and 'Project build ended successfully'.

Criação de programa:

- Inicie a execução do **Pelles C IDE**.
- No menu, selecione **File** → **New** → **Project** → **Win32 Console program**.
- Dê um nome ao projeto e clique **OK**.
- Pressione **Ctrl-N** para criar o arquivo.
- Digite o código do programa.
- Pressione **Ctrl-S** para salvar.
- Pressione **Ctrl-F5** para executar.

Esse compilador é de uso livre e está disponível para *download* na página da disciplina!



Introdução

Tipos de dados

definem conjuntos de **valores** e respectivas **operações** que podem ser feitas com eles.

Tipo básico	Bytes	Escala
<code>char</code>	1	$-128 \dots +127$
<code>int</code>	4	$-2\,147\,483\,648 \dots +2\,147\,483\,647$
<code>float</code>	4	$3.4e-38 \dots 3.4e+38$ (absoluto)
<code>double</code>	8	$1.7e-308 \dots 1.7e+308$ (absoluto)
<code>void</code>	0	vazia

Tipo modificado	Bytes	Escala
<code>unsigned char</code>	1	$0 \dots 255$
<code>unsigned int</code>	4	$0 \dots 4294967295$
<code>short int</code>	2	$-32767 \dots +32767$
<code>long long int</code>	8	$-9223372036854775808 \dots +9223372036854775807$
<code>unsigned short int</code>	2	$0 \dots 65535$
<code>unsigned long long int</code>	8	$0 \dots 18446744073709551615$



Introdução

Tipos e constantes:

- **char** (caractere) :
 - 'a', '2', '@', '\n', ...
- **int** (número inteiro) :
 - Decimal: 1, 234, 56, ...
 - Octal: 05, 067, 012, ...
 - Hexadecimal: 0x1, 0x7F, 0x8A, ...
- **double** (número real) :
 - Notação comum: 0.1, 2.34, 5.0, ...
 - Notação científica: 5.67e2, 1e3, , 0.1e4, ...
- Não há o tipo **string** (cadeia de caracteres) em C, mas há constantes desse tipo:
 - "b", "1", "Ana", ...

Em C, não há distinção entre caracteres e seus respectivos códigos ASCII!



Introdução

Tipos e variáveis:

- Toda variável deve ser **identificada** e **declarada** antes de ser usada.
- Um **identificador** deve iniciar com uma **letra** (seguida de letras, dígitos e sublinhas).
- A **declaração** consiste de um **tipo** seguido de um **identificador**.
- O tipo determina a **quantidade de memória** que deve ser alocada para a variável.
- O identificador possibilita **citar** a variável no bloco no qual ela é declarada.
- Uma variável pode ser **iniciada** ao ser declarada (com o operador de atribuição **=**).

Exemplo 2. Declaração de variável

```
char tecla = 'A', opcao;  
int x, y = 3, z;  
float comissao = 0.10, desconto, salario;
```

Em C, não é possível declarar uma variável simples do tipo **void**!



Introdução

Operadores aritméticos

são usados para representar operações aritméticas:

- **Soma**.....: **+**
 - **Subtração**...: **-**
 - **Produto**: *****
 - **Divisão**.....: **/**
 - **Resto**.....: **%**
- (o resultado é inteiro apenas quando os dois valores são inteiros)
- (produz o resto da divisão entre dois números inteiros)

Exemplo 3. Resultado dos operadores aritméticos

- **1 + 2** \Rightarrow **3**
- **8.0 - 4** \Rightarrow **4.0**
- **3 * 7** \Rightarrow **21**
- **7 / 2** \Rightarrow **3**
- **7 / 2.0** \Rightarrow **3.5**
- **7 % 2** \Rightarrow **1**
- **7.0 % 2** \Rightarrow **erro**



Introdução

O operador `sizeof`

informa a quantidade de memória (em bytes) usada por um tipo de dados. Pode ser aplicado a **tipo**, **constante**, **variável** ou **expressão**.

Exemplo 4. Tamanho de um tipo

```
#include <stdio.h>

int main(void) {
    char v = 'A';
    printf("%d\n", sizeof(int));
    printf("%d\n", sizeof(1.2));
    printf("%d\n", sizeof(v));
    printf("%d\n", sizeof(v+1));
    return 0;
}
```



Introdução

A função `scanf()`

lê dados do teclado (**entrada padrão**) e os armazena em variáveis.

```
scanf("formatação", end1, end2, ..., endn);
```

endereços de variáveis

especificadores de formato (%): indicam os tipos e a ordem em que os dados devem ser digitados.

demais caracteres: devem ser digitados pelo usuário.

Exemplo 5. Entrada formatada

```
int idade;  
char sexo;  
scanf("%d %c", &idade, &sexo);
```

A função `scanf()` está declarada em `stdio.h` (*standard input/output header*).



Introdução

Especificador de formato

determina o tipo de dado que será lido do teclado ou exibido no vídeo.

Especificador	Representa
%c	caractere
%o, %d, %x, %X	número inteiro em octal, decimal ou hexadecimal
%u	número inteiro em base decimal sem sinal
%hd	número inteiro curto em base decimal
%lld	número inteiro longo longo em base decimal
%f	número real de precisão simples ou dupla
%s	cadeia de caracteres (string)
%%	único sinal de porcentagem

Esses são os especificadores mais usados; porém, há vários outros!



Introdução

A função `printf()`

exibe dados no vídeo (saída padrão).

```
printf("formatação", val1, val2, ..., valn);
```

valores

- especificadores de formato (%): são substituídos pelos valores correspondentes.
- caracteres de controle (\): produzem efeitos especiais (e.g., som e mudança de linha).
- demaís caracteres: são exibidos literalmente no vídeo do computador.

Exemplo 6. Saída formatada

```
int idade;  
char sexo;  
printf("%d %c\n", idade, sexo);
```

A função `printf()` está declarada em `stdio.h` (*standard input/output header*).



Introdução

Caractere de controle

representa um caractere especial da tabela ASCII.

Caractere de controle	Efeito na saída de dados
<code>\a</code>	soa o alarme do microcomputador
<code>\b</code>	o cursor retrocede à coluna anterior
<code>\f</code>	alimenta página na impressora
<code>\n</code>	o cursor avança para uma nova linha
<code>\r</code>	o cursor retrocede para a primeira coluna da linha
<code>\t</code>	o cursor avança para próxima marca de tabulação
<code>\"</code>	exibe uma única aspa
<code>\'</code>	exibe um único apóstrofo
<code>\\</code>	exibe uma única barra invertida
<code>\0</code>	indica o final de uma cadeia de caracteres

Esses são os caracteres de controle mais usados; porém, há vários outros!



Introdução

Formatação de campos para exibição

- **Preenchimento com espaços**...: `% tamanho_do_campo d`
- **Preenchimento com zeros**.....: `%0 tamanho_do_campo d`
- **Número de casas decimais**.....: `% tamanho_do_campo . número_de_casas f`

Exemplo 7. Formatação de campos

```
#include <stdio.h>

int main(void) {
    int a = 678;
    float b = 12.3416;
    printf("%5d\n", a);           // □□678
    printf("%06d\n", a);         // 000678
    printf("%7.3f\n", b);        // □12.342
    return 0;
}
```



Introdução

Exercício 1. IMC

Execute o programa `obeso.c`, usando o compilador *Pelles C*.

Exercício 2. Erro

Qual é o erro no programa a seguir?

```
#include <stdio.h>
int main(void) { printf("%d\n", 0678); return 0; }
```

Exercício 3. Bytes

Crie um programa para informar quantos bytes são necessários para representar os resultados das expressões $7/2$ e $7/2.0$, respectivamente.

Exercício 4. Conversão

Dado um número inteiro em base 10, exibir o valor correspondente em base 16.

Exercício 5. ASCII

Dado um caractere, exiba o código ASCII correspondente.



Introdução

Exercício 6. Formatação

Exiba o texto "`c:\backup\texto\novo\arq.txt`" no vídeo.

Exercício 7. Média

Dadas as duas notas de prova de um aluno, informe a sua média.

Exercício 8. Consumo

Dada uma distância percorrida em quilômetros, e o total de litros de combustível gasto para percorrê-la, informe o consumo médio do veículo.

Exercício 9. Temperatura

Dada uma temperatura em graus Fahrenheit (°F), informe a temperatura correspondente em graus Celsius (°C). [**Dica:** $C = (F - 32) * (5/9)$].

Exercício 10. Distância

Dadas as coordenadas (**xp**, **yp**) e (**xq**, **yq**), de dois pontos **P** e **Q** no plano cartesiano, exiba a distância entre eles. [**Dica:** use o Teorema de Pitágoras].

Fim

