

# Lógica de Programação

CSTSI CEFET-RS

# Instruções que o computador executa

- Instrução para **enviar** uma informação para o dispositivo de **saída** (escrever no vídeo).
- Instrução para **armazenar** uma informação na **memória** principal.
- Instrução para **obter** um dado a partir do dispositivo de **entrada** (ler o teclado).
- Instrução para **obter** o valor de **expressões** aritméticas (calcular).

# Primeiro exemplo

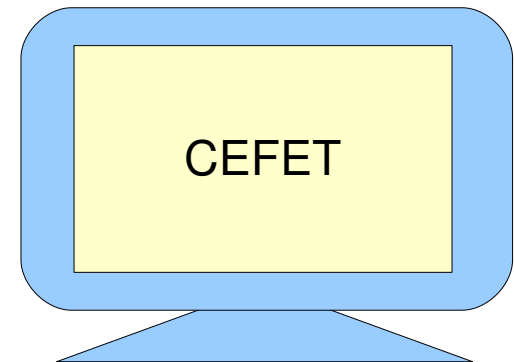
Escrever um algoritmo para exibir na tela do computador a palavra CEFET.

## Português estruturado

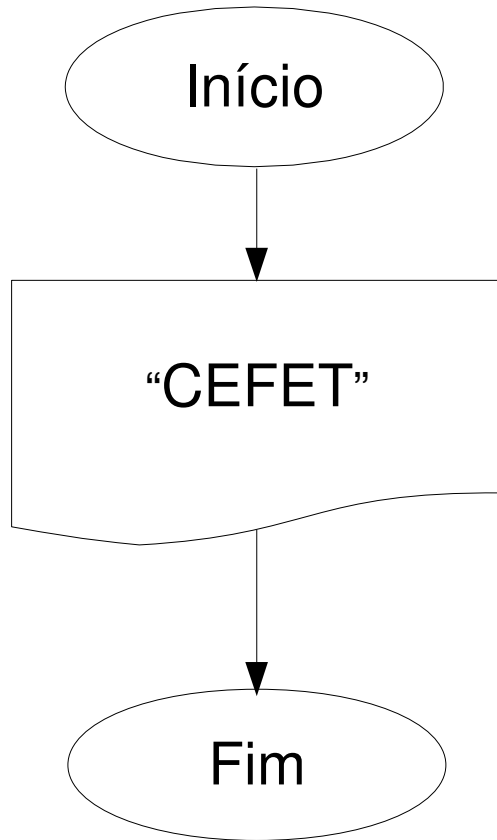
início

    Escreva "CEFET"

fim



## Fluxograma



## Diagrama de Chapin



# E como fica em C?

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

```
main( )
```

```
{
```

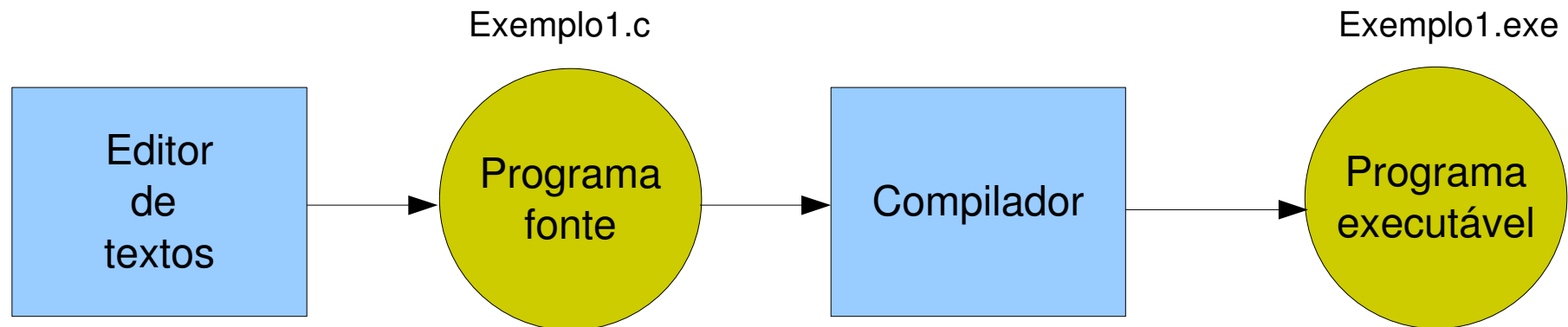
```
printf( "CEFET" );
```

```
}
```

Como executar o programa no computador?

# Como executar o programa no computador?

1. Digitar o programa em um editor de textos (programa fonte) .
2. Compilar o programa (traduzir o programa para linguagem de máquina gerando um executável).
3. Executar o programa.



# Compiladores

Existem vários compiladores para a linguagem C.

Dev C++ (<http://www.bloodshed.net/devcpp.html>)

Turbo C

Turbo C++

Builder C++

Microsoft C

gcc

Visual C++

# Dev C++

Para editar o programa:

**Arquivo --> Novo --> Arquivo Fonte**

Para gravar o programa:

**Salvar --> a) Fornecer o nome do arquivo**

**b) Alterar o tipo do arquivo para:**

**C Source files (\*.c)**

Para compilar e executar o programa:

**Executar --> Compilar & Executar (F9)**



Inserir as linhas que estão em vermelho para evitar que a janela se feche após a impressão da palavra CEFET.

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

```
main( )  
{  
printf( "CEFET" );  
system( "PAUSE" );  
}
```

# Linguagem C

- Desenvolvida por Dennis Ritchie na década de 70.
- Linguagem de uso geral.
- Foi utilizada no desenvolvimento do sistema operacional UNIX.

## **Características de um programa em C**

- Diferencia letras maiúsculas de minúsculas.
- Os comandos são terminados por ;
- O formato é livre.

função “principal”  
indica onde o programa  
inicia

inclui o arquivo stdio.h

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

delimitador de bloco

string

```
main( )
```

```
{
```

```
printf( "CEFET" );
```

fim de comando

comando de saída

```
system( "PAUSE" );
```

```
}
```

delimitador de bloco

executa uma comando  
do sistema (pausa)

stdio.h : Arquivo de *header* (***standard input output***)

# Formato livre

## Exemplo 1

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

main () {printf("CEFET");system("PAUSE");}
```

---

## Exemplo 2

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

main      (      ) {
printf
(
      "CEFET"
)
;
      system
(
      "PAUSE"      );
}
```

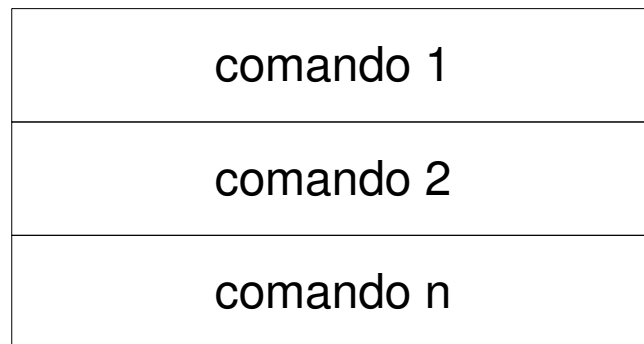
# Controle do fluxo de execução

- Estruturas básicas de controle do fluxo de instruções de um algoritmo.
- De acordo com a forma com que este controle é feito, estas estruturas são classificadas em:
  - Estruturas **seqüenciais**.
  - Estruturas de **seleção**.
  - Estruturas de **repetição**.

# Estruturas seqüenciais

- Comandos são executados em uma seqüência pré-estabelecida.
- Cada comando é executado somente após a finalização do comando anterior.
- A estrutura seqüencial é caracterizada por um único fluxo de execução.

Diagrama de Chapin



Português estruturado

```
início  
    comando 1  
    comando 2  
    comando n  
fim
```

# Exemplo 2

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

main( )
{
printf( "Lógica" );
printf( "de" );
printf( "Programação" );
system( "pause" );
}
```

## printf

Envia dados ao dispositivo de saída (tela).

Equivale ao comando **Escreva**

## Chapin

Escreva "Lógica"
Escreva "de"
Escreva "Programação"

# Exemplo 3

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

main( )
{
printf( "Lógica\n" );
printf( "de\n" );
printf( "Programação\n" );
system( "pause" );
}
```

## Chapin

Escreva "Lógica"
Escreva "de"
Escreva "Programação"

**\n**

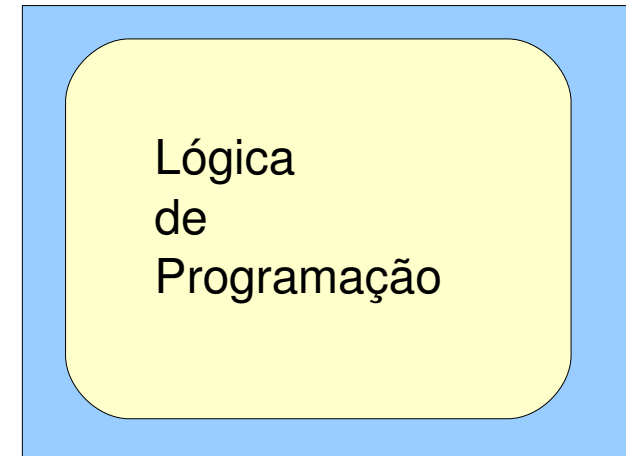
Código que significa nova linha



## Exemplos 4 e 5

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

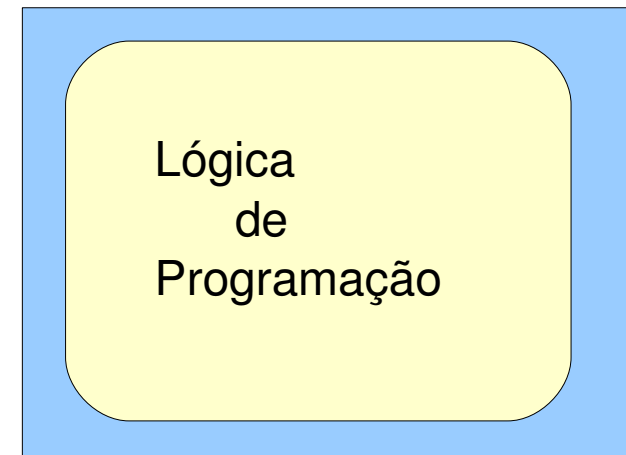
main()
{
printf("Lógica\n");
printf("de\n");
printf("Programação\n");
system("pause");
}
```



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

OBS: O espaço só é impresso **dentro** das “aspas”

```
main()
{
printf("Lógica\n");
printf("    de\n");
printf("Programação\n");
system("pause");
}
```



# Exercícios

1) Escreva exatamente o que cada trecho de programa imprime. Logo após complete, digite, compile e execute cada programa para conferir a sua resposta

(a)   ...

```
printf( "*" );  
printf( "   ***" );  
printf( "***\n***  " );  
printf( "***" );  
printf( "*\n" );  
...
```

(b)   ...

```
printf( "*" );  
printf( "   ***\n" );  
printf( "*****\n  " );  
printf( "*****" );  
printf( "   **\n" );  
...
```

# Exercícios

- 2) Escreva um programa em C para exibir Lógica de Programação exatamente como a tela abaixo.

