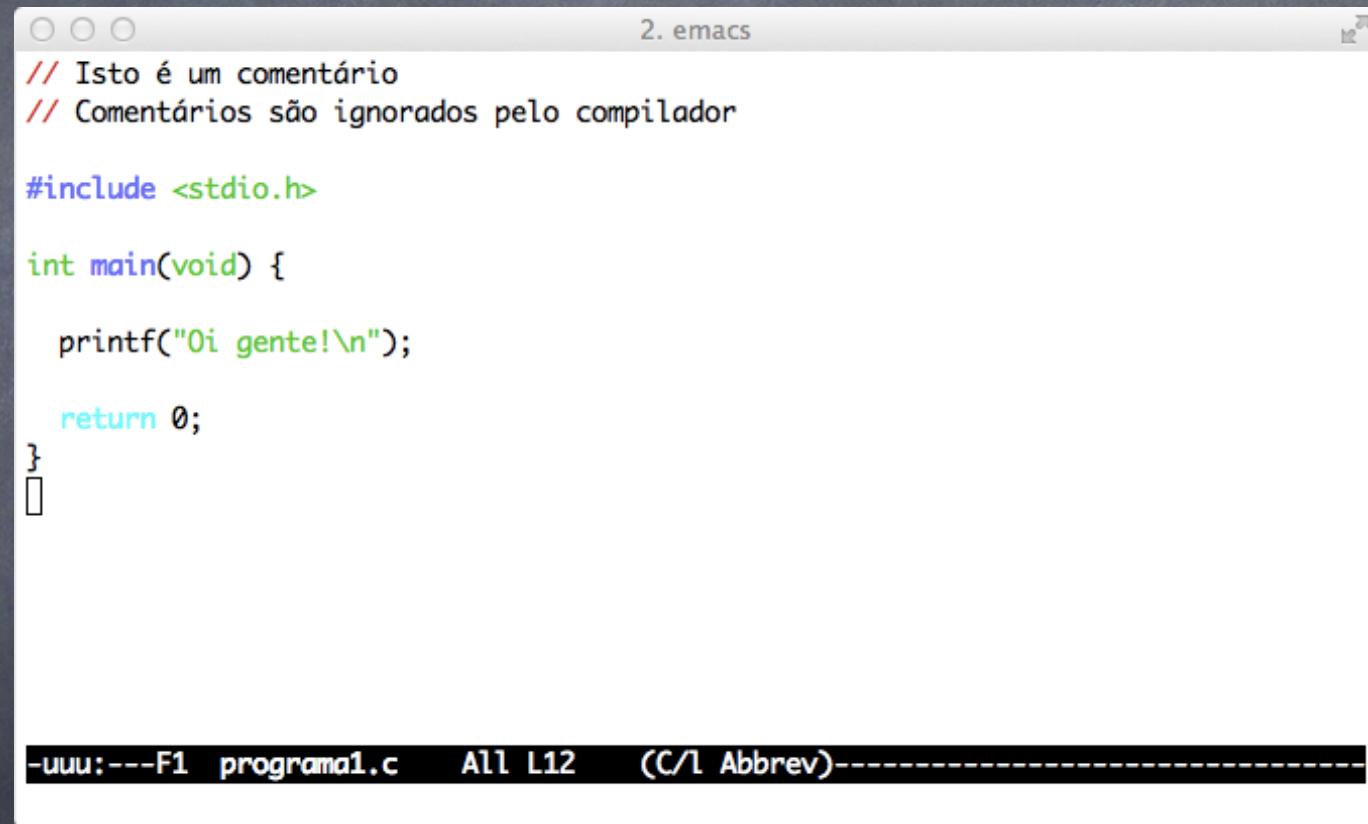


# Como executar um programa

- ➊ Os seguintes passos serão utilizados para resolver um problema usando o computador:
  1. entender bem o problema em questão
  2. encontrar a solução mais eficiente
  3. se necessário, escrever o algoritmo em português, para organizar as idéias
  4. escrever o programa (na linguagem C), guardando-o em um arquivo de computador (chamado **arquivo-fonte**)
  5. **compilar** o programa armazenado no arquivo-fonte. Um **compilador** é um programa que traduz a linguagem C para uma linguagem que o computador sabe executar (chamada linguagem de máquina)

# Um primeiro programa



The image shows a screenshot of an Emacs window titled "2. emacs". The buffer contains the following C code:

```
// Isto é um comentário
// Comentários são ignorados pelo compilador

#include <stdio.h>

int main(void) {
    printf("Olá gente!\n");
    return 0;
}
```

The code is syntax-highlighted, with comments in red, strings in green, and keywords in blue. The window has a standard OS X-style title bar and scroll bars.

At the bottom of the window, there is a status bar with the following information:

-uuu:---F1 programma1.c All L12 (C/l Abbrev)-----

# Comentários sobre o primeiro programa

- Programa usa um único tipo de comando: `printf`
- Muitos detalhes!
- Note o uso dos caracteres '//' : marca o início de um comentário.
- Um comentário inicia com os caracteres '//' e se estende até o final da linha corrente.
- Comentários são ignorados na execução do programa, servem para documentar o programa.

# Executando o primeiro programa

```
// Isto é um comentário  
// Comentários são ignorados pelo compilador  
  
#include <stdio.h>  
  
int main(void) {  
    printf("Oi gente!\n");  
  
    return 0;  
}  
  
-uuu:---F1  programa1.c  All L12  (C/l Abbrev)
```

```
Ricardos-MacBook-Air-2:src ricardoanido$ gcc -o programa1 programa1.c  
Ricardos-MacBook-Air-2:src ricardoanido$ ./programa1  
Oi gente!  
Ricardos-MacBook-Air-2:src ricardoanido$ 
```

# Mais sobre o comando printf

- Comando **printf** é utilizado para imprimir os resultados do programa.
- A cadeia a ser impressa deve ser colocada entre aspas duplas:

```
printf("uma cadeia de caracteres\n");
```

- \n faz uma nova linha ser impressa

```
printf("Uma linha\nE outra linha\n");
```

# Mais sobre o comando printf

- A cadeia do comando printf é chamada da “cadeia formatadora”, porque ela também serve para **formatar** (indicar o formato) de valores a serem impressos.
- Por exemplo, usamos %d para indicar que um valor inteiro deve ser impresso; o valor a ser impresso após a cadeia formatadora, separado por uma vírgula:

```
printf("resultado: %d\n", 100+100);
```

# Mais sobre o comando printf

- Podemos usar a cadeia formatadora de um comando printf para imprimir mais de um valor. Nesse caso, a lista de valores deve vir após a cadeia formatadora, cada valor (ou expressão) separado por uma vírgula:

```
printf("%d + %d é igual a %d\n", 10, 10, 10+10);
```

- Note que o número de %d na cadeia formatadora deve ser igual ao número de valores da lista de valores.

# Variáveis

- ⦿ Em programação, uma variável é um nome associado a um valor
- ⦿ Uma variável é composta por
  - ⦿ um nome
  - ⦿ um tipo
  - ⦿ um valor

# Nomes de Variáveis

- Devem ser formados por combinações de letras maiúsculas (A-Z), minúsculas (a-z), dígitos (0-9) e o caractere sublinhado ( \_ ).
- Não podem começar com dígitos.
- Não podem conter acentuações.
- Não podem ser uma palavra reservada da linguagem de programação (uma variável não pode se chamar por exemplo printf).

# Nomes de Variáveis

- ➊ Exemplos
  - ➋ nomes corretos: var1, var2, media, ValorMaximo
  - ➋ nomes incorretos: Média, printf, 2var
- ➋ **Atenção:** letras maiúsculas e minúsculas fazem diferença
  - ➋ variável **maxVal** é diferente da variável **MaxVal**

# Criando variáveis em C

- Para criar uma variável em C, é necessário **declarar** a variável, indicando o seu **nome** e o seu **tipo**
- A variável criada somente pode armazenar **valores** do **tipo** declarado

## ● Exemplos

```
int nota1;  
int nota2;  
float media;  
int a,b,c;
```

# Comando de atribuição

# Comando de atribuição

- O comando de atribuição é utilizado para armazenar um valor em uma variável
- O formato geral do comando é:  
`nome_de_variavel = expressão ;`
- A partir da atribuição, a variável contém o valor atribuído e pode ser utilizada em qualquer expressão (inclusive em outras atribuições).

# Comando de atribuição

## Exemplos

```
int x,y,z;  
  
x = -10;  
y = 20;  
z = x + y;  
z = z + 1;  
printf("%d %d\n",x,y);
```

```
int a,b;  
float media;  
  
a = 89;  
b = 67;  
media = (a + b)/2.0;  
printf("%f\n",media);
```

# Operações Aritméticas

## • Operadores aritméticos:

operador	operação
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão

# Operadores aritméticos

## Exemplos

```
a+b      // adição de a e b  
b*2      // b vezes 2  
conta % 5 // resto da divisão de conta por 5
```

# Comando de leitura

# Comando de leitura

- O comando de leitura permite que o programa interaja com o “mundo externo”.
- É a forma de fazer **entrada de dados** para o programa.
- O formato geral do comando é:  
`scanf("cadeia formatadora", &var1, &var2, ...);`
- O comando de leitura lê da **entrada padrão** associada ao programa (normalmente o teclado).

# Comando de leitura

- Exemplo:

```
scanf( "%d %d", &nota1, &nota2);
```

- O que ocorre na execução:

- o programa espera que o usuário digite dois inteiros seguido de "enter". O primeiro inteiro lido é atribuído à variável inteira nota1, o segundo inteiro lido é atribuído à variável inteira nota2.

# Comando de leitura

- Podemos também ler variáveis de tipo real, usando o indicativo `%f` na cadeia formatadora. As variáveis de leitura devem então ser do tipo `float`.

- Exemplo

```
float x, y;
```

```
...
```

```
scanf ("%f %f", &x, &y);
```

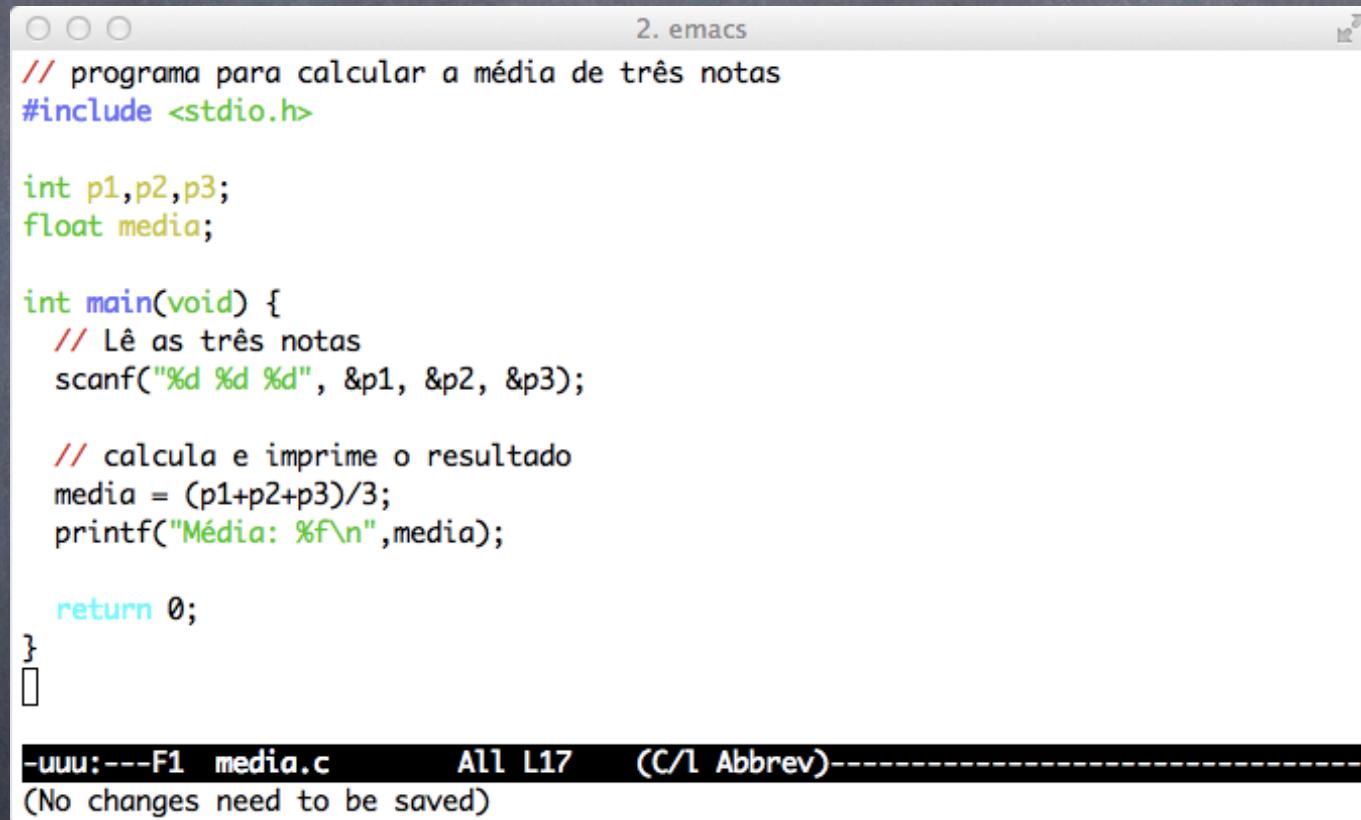
# Tarefa 1

## Média

# Primeira Tarefa

- ➊ Sua professora está cansada de calcular as médias dos alunos da turma e pediu que você escreva um programa que:
  - ➋ leia uma linha com três inteiros, correspondendo às notas de três provas, P1, P2 e P3.
  - ➋ calcule a média das notas das três provas e imprima uma linha contendo o resultado.

# Primeira Tarefa



The screenshot shows a terminal window titled "2. emacs" displaying a C program. The code calculates the average of three grades (p1, p2, p3) and prints the result. The terminal status bar at the bottom indicates the file is "media.c", has 17 lines, and no changes need to be saved.

```
// programa para calcular a média de três notas
#include <stdio.h>

int p1,p2,p3;
float media;

int main(void) {
    // Lê as três notas
    scanf("%d %d %d", &p1, &p2, &p3);

    // calcula e imprime o resultado
    media = (p1+p2+p3)/3;
    printf("Média: %f\n",media);

    return 0;
}

-uuu:---F1  media.c      All L17  (C/l Abbrev)-----
(No changes need to be saved)
```