Introdução à Programação C

Fabio Mascarenhas - 2014.2

http://www.dcc.ufrj.br/~fabiom/introc

Recapitulando

- Um programa C é uma sequência de diretivas (#include e #define), declarações de funções auxiliares, e uma função principal *main*

- O corpo de uma função em C é uma sequência de declarações de variáveis seguida por uma sequência de comandos (atribuições, chamadas de função, if, while, for)

- Declarações de variáveis e comandos simples devem sempre terminar com ; (ponto e vírgula)

- Em uma declaração de variável damos o *tipo* da variável primeiro, depois uma lista de variáveis que estamos declarando

Exemplo

```c
#include <stdio.h>

/* Função auxiliar */
static double converte(double c) {
    double f;
    f = 1.8 * c + 32;
    return f;
}

/* Função principal */
int main()
{
    double t1, t2;
    puts("Digite a temperatura em celsius: ");
    scanf("%lf", &t1);
    t2 = converte(t1);
    printf("Temperatura em Farenheit: %lf\n", t2);
    return 0;
}
```

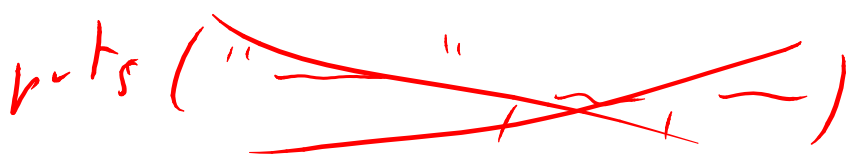
Entrada e saída no console

- Vamos fazer toda a entrada e saída no console dentro da função principal, como no programa exemplo

- Existem três funções básicas de entrada e saída: puts, printf e scanf

- A função puts é como print em Python: recebe uma mensagem e mostra ela no console, seguida de uma quebra de linha

```
puts(“Digite a temperatura em celsius: ");
```



A função printf

- Quando temos mais do que uma simples mensagem para mostrar usamos a função printf, que formata uma mensagem contendo várias partes antes de mostrá-la no console

- Chamamos printf passando um *modelo* para a mensagem, seguidos dos *valores* que queremos plugar nesse modelo

- O modelo inclui *códigos de formato* que indicam os pontos onde os valores vão ser “plugados”, e com qual formato

```
printf(“Temperatura em Farenheit: %lf\n", t2);
```

códigos de formato

modelo

valor

print "%.1f C is %.1f F" % (t1, t2) ≡ "23 C is 73 F"

Códigos de formato

- Os códigos básicos de formato são %c, %d, %lf e %s, correspondendo respectivamente a: caracteres (char), inteiros (int), reais (double) e cadeias de caracteres

- Os códigos %d e %lf podem também incluir uma especificação de *tamanho*:

  - %5d imprime o inteiro fazendo ele ocupar pelo menos cinco caracteres, preenchendo o que falta com espaços à esquerda, já %.5d imprime o inteiro fazendo ele ocupar pelo menos cinco caracteres, preenchendo o que falta com zeros à esquerda

  - %.2lf imprime o número real fazendo ele ter exatamente dois dígitos após o ponto decimal, preenchendo o que falta com zeros à direita, já %8.2lf imprime o número real com exatamente dois dígitos após o ponto, e o número todo ocupando pelo menos oito caracteres, preenchendo o que falta com espaços à esquerda

Entrando valores com scanf

- A função scanf espera o usuário entrar um ou mais valores o console, e coloca eles em variáveis

- Como printf, também passamos para scanf um modelo, mas esse modelo diz que tipos de valores queremos ler, e tem apenas códigos de formato simples separados por espaços

```
scanf("%lf", &t1);
```

- Cada variável passada depois do modelo tem que ser prefixada com &

```
scanf("%d %lf %d", &a, &b, &c);
```

Execução passo a passo

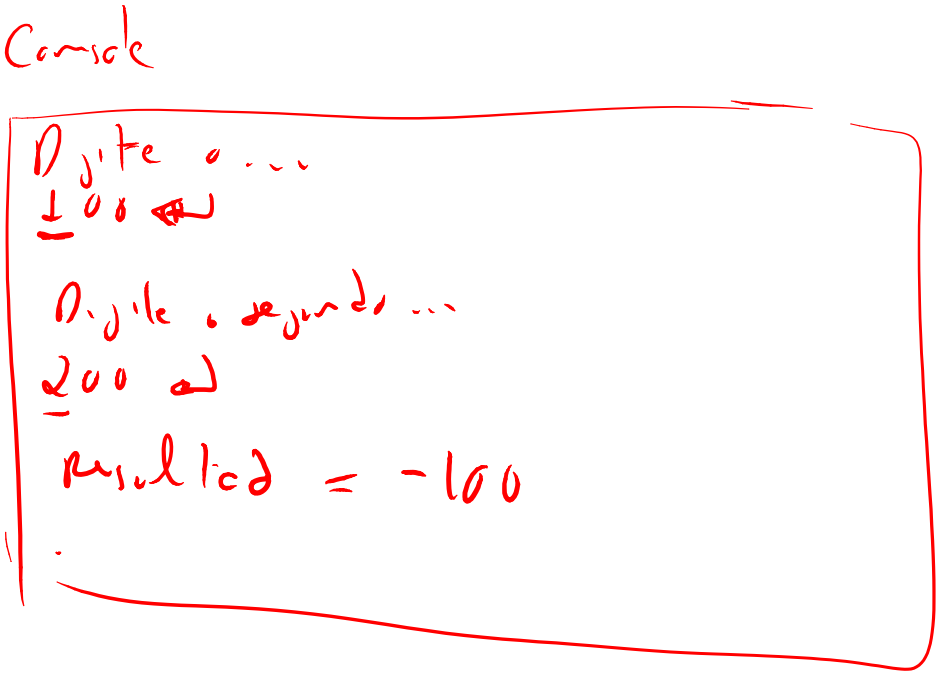
- No ambiente de desenvolvimento podemos executar um programa passo a passo, o que é uma importante ferramenta de aprendizado

- Mas também é útil saber fazer isso com papel e caneta, para algoritmos simples

```
#include <stdio.h>

int main()
{
    int numero1, numero2, diferenca;
    puts(“Digite o primeiro numero inteiro: ");
    scanf("%d", &numero1);
    puts(“Digite o segundo numero inteiro: ");
    scanf("%d", &numero2);
    diferença = numero1 – numero2;
    printf(“Resultado da diferença = %d\n", diferenca);
    return 0;
}
```

Passo a passo do exemplo




```c
#include <stdio.h>

int main()
{
    int numero1, numero2, diferenca;
    puts("Digite o primeiro numero inteiro: ");
    scanf("%d", &numero1);
    puts("Digite o segundo numero inteiro: ");
    scanf("%d", &numero2);
    diferença = numero1 - numero2;
    printf("Resultado da diferença = %d\n", diferenca);
    return 0;
}
```

Exercício

só função main!

- Escreva um programa C que leia um valor em segundos e o imprima no formato horas:minutos:segundos, em que minutos e segundos sempre têm dois dígitos, preenchidos com um 0 à esquerda se preciso (por exemplo, 4322 segundos é 1:12:02)

Entrada: 8000 | 3600

Saída: 2:13:20 | 1:00:00

%..2d → dois dígitos, 0 à esquerda