**Computação para Informática - Prof. Adriano Joaquim de Oliveira Cruz**
**Segunda Aula Prática**

---

# 1 Introdução

Para compilar os programas use o seguinte comando:

## gcc -o programa programa.c -Wall -ansi

programa é substituído pelo nome que você escolheu para o seu programa.

O objetivo desta aula é praticar algoritmos e começar a escrever programas em `C`. Para isto vamos executar três tipos de tarefas:

1. exercitar o uso de variáveis de vários tipos;
2. iniciar o estudo de comandos de entrada e saída simples;
3. testar alguns algoritmos escritos em pseudo-código.

**Sugestões:**

- Os seus programas devem começar sempre com os comentários mostrados na listagem 1. Claro que colocando informações relevantes, por exemplo a data correta.
- Crie um arquivo com este conteúdo. Toda vez que for escrever um programa abra este arquivo e depois logo em seguida salve-o com o nome desejado. Deste modo você economizará o tempo de escrever estes comandos.

Listagem 1: Padrão para programas 1.

```
/*
    Programa:  NomeDoArquivoFonte.c
    Autor:     Aluno Programador Brilhante
    Data:      dd/mm/aaaa
    Descrição: Este programa faz algo muito importante.
*/
#include<stdio.h>
int main (void) {
    /* Aqui vao as declaracoes das variaveis */

    /* Aqui vao os comandos do programa */

    return 0;
}
```

## 2 Declaração de variáveis

Todas as variáveis em `C` devem ser definidas (declaradas) antes de serem usadas. Esta definição deve ser dada no início do programa, segundo o padrão ANSI.

Os dados em `C` podem assumir cinco tipos básicos que são os seguintes:

**char**: O valor armazenado é um caractere.

**int**: O valor armazenado é um número inteiro.

**float**: Número em ponto flutuante de precisão simples, normalmente 32 bits. São conhecidos como números reais.

**double**: Número em ponto flutuante de precisão dupla, com isto a precisão e as vezes a excursão dos números aumenta. Este tipo é armazenado em 64 bits.

**void**: Este tipo serve para indicar que um resultado não tem um tipo definido. Uma das aplicações deste tipo em `C` é criar um tipo vazio que pode posteriormente ser modificado para um dos tipos anteriores.

---

**Exercício 1:** Escreva o programa mostrado na listagem 2 e verifique o seu funcionamento. Observe que todas as variáveis foram definidas logo no início do programa. Não se preocupe, a função `printf` será o próximo item a ser apresentado.

Listagem 2: Definição de variáveis 1.

```
/*
    Programa:  definicao.c
    Autor:     Aluno Programador Brilhante
    Data:      dd/mm/aaaa
    Descrição: Este programa mostra exemplo de definicao de variaveis.
*/
#include<stdio.h>
int main (void) {
     int i;
     float r;
     double dr;
     char c;

     i = 10;
     r = 1.0;
     dr = 3e3;
     c = 'a';

     printf("i = %d\n", i);
     printf("r = %f\n", r);
     printf("df = %f\n", dr);
     printf("c = %c\n", c);

     return 0;
}
```

# 3 A Função `printf`

A função `printf` faz com que dados sejam escritos na saída padrão, que normalmente é a tela do computador. O protótipo da função é:

```
int printf("formato", var1, var2, ...);
```

onde os argumentos `var1, var2, ...` são impressos de acordo com o formato indicado pela cadeia de caracteres que compõe `formato`.

Um exemplo simples pode tornar a explicação mais clara. O programa 3 imprime o valor das variáveis `dia, mes, ano`. Detalhando o comando `printf` temos primeiro entre aspas o seguinte texto:

```
Estamos na data: dia = %d, mes = %d e ano = %d.
```

Este texto contém as indicações do que e como deve ser impresso. Tudo que tiver precedido pelo caractere % é indicação do formato de saída dos dados. O resto vai direto para a saída. Por exemplo, neste texto `%d` indica que uma variavel inteira vai ser impressa. Como temos três destes formatos devemos ter de imprimir três variáveis. Observe então que o comando `printf` termina com a lista das variáveis a serem impressas.

Listagem 3: Exemplo de impressão de resultados

```
#include <stdio.h>
int main (void)  {
    int ano = 1997, dia = 29, mes = 12;

    /* Imprime o valor do ano */
    printf("Estamos na data: dia = %d, mes = %d e ano = %d\n", dia, mes, ano);
    return 0;
}
```

A tabela 1 mostra alguns dos códigos usados para **saída de dados**. A tabela completa é assunto para outra aula prática.

| Código | Comentário |
|---|---|
| `%c` | Caracter simples |
| `%s` | Cadeia Caracteres |
| `%d` ou `%i` | Inteiro (`int`) decimal com sinal |
| `%f` | Real em ponto flutuante |
| `%%` | Imprime o caractere % |

Tabela 1: Alguns códigos de conversão para escrita de dados.

---

**Exercício 2:** Escreva o programa mostrado na Listagem 3 e verifique o seu funcionamento.

---

**Exercício 3:** Escreva um programa que defina variáveis dos tipos `float` atribua valores a estas variáveis e as imprima usando o formato normal.

---

# 4 A função scanf

---

**Exercício 4:** Escreva um programa que calcule a nota de um aluno de uma disciplina cuja fórmula é a seguinte:

$$notaFinal = 0.8 * prova + 0.2 * \frac{\sum_{i=0}^{n-1} teste_i}{n} \quad (1)$$

Considere que o número de testes é igual a 3.

Exemplos de saída:

```
Prova:  8.0
Teste 1:  8.0
Teste 2:  10.0
Teste 3:  3.0
Nota final 7.8
```

A listagem 4 mostra como este programa poderia ser escrito.

Listagem 4: Exemplo de leitura de dados.

```c
#include<stdio.h>
int main (void) {
    float prova;
    float teste1, teste2, teste3;
    float notaFinal;

    printf("Prova: ");
    scanf("%f", &prova);
    printf("Teste 1: ");
    scanf("%f", &teste1);
    printf("Teste 2: ");
    scanf("%f", &teste2);
    printf("Teste 3: ");
    scanf("%f", &teste3);
    notaFinal = 0.8 * prova + 0.2 * (teste1 + teste2 + teste3) / 3 ;
    printf("Nota final %0.2f\n", notaFinal);

    return 0;
}
```

A tabela 2 mostra alguns dos códigos usados para **entrada de dados**. A tabela completa é assunto para outra aula prática.

| Código | Comentário |
|---|---|
| `%c` | Caracter simples |
| `%s` | Cadeia Caracteres |
| `%d` ou `%i` | Inteiro (`int`) decimal com sinal |
| `%f` | `float`: Real em ponto flutuante |
| `%lf` | `double`: Real em ponto flutuante |
| `%%` | Imprime o caractere % |

Tabela 2: Alguns códigos de conversão para entrada de dados.

# 5 Desafios

---

**Exercício 5:** Escreva um programa que imprima a soma de todos os números inteiros entre 0 e $N$. Leia o valor de $N$ do teclado. **Não use comando de repetição.**

---

**Exercício 6:** Considere o algoritmo 1. Este algoritmo convertido para programa fica da forma mostrada na listagem 5. Repare que faltam alguns pedaços. Olhe o algoritmo e complete o programa e teste o seu funcionamento.

---

**inicio**
  **ler** *notaAluno*
  $mediaAluno \leftarrow notaAluno$
  **ler** *notaAluno*
  $mediaAluno \leftarrow mediaAluno + notaAluno$
  **ler** *notaAluno*
  $mediaAluno \leftarrow mediaAluno + notaAluno$
  $mediaAluno \leftarrow mediaAluno/3$
  **imprimir** "A média é ", *mediaAluno*
  **se** $mediaAluno \geq 5.0$ **então**
    **imprimir** "Aprovado"
  **senão**
    **imprimir** "Reprovado "
  **fim se**
**fin**

---

**Exercício 7:** Considere o algoritmo 2. Este algoritmo convertido para programa fica da forma mostrada na listagem 6. Repare que faltam alguns pedaços. Olhe e complete o programa e teste o seu funcionamento.

Listagem 5: Calcula a media de três provas.

```c
/*
 * Programa: mediaTresNotas.c
 * Autor: Adriano Cruz
 * Descricao: Le tres notas de um aluno, calcula sua media
 *            e informa se ele foi aprovado ou não.
 * Data: 2013/08/27
 */
#include<stdio.h>
int main (void) {
    float notaAluno;
    float mediaAluno;

     /* OOPS aqui falta codigo. */

    if (mediaAluno >= 5.0) {
     printf("Aprovado\n");
    }
    else {
     printf("Reprovado");
    }

    return 0;
}
```

---

**inicio**
  $mediaAluno \leftarrow 0$
  **imprimir** "Quantas provas?"
  **ler** $quantProvas$
  $notasLidas \leftarrow 0$
  **enquanto** $notasLidas < quantProvas$ **faça**
    **ler** $notaAluno$
    $mediaAluno \leftarrow mediaAluno + notaAluno$
    $notasLidas \leftarrow notasLidas + 1$
  **fim enqto**
  $mediaAluno \leftarrow mediaAluno/quantProvas$
  **imprimir** "A média é ", $mediaAluno$
  **se** $mediaAluno \geq 5.0$ **então**
    **imprimir** "Aprovado"
  **senão**
    **imprimir** "Reprovado "
  **fim se**
**fin**

---

Listagem 6: Calcula a media de provas.

```c
/*
 * Programa: mediaNotas.c
 * Autor: Adriano Cruz
 * Descricao: Calcula media de aluno e informa se ele foi aprovado ou não.
 *            O numero de provas deve ser lido do teclado.
 * Data: 2013/08/27
 */
#include<stdio.h>
int main (void) {
    float notaAluno;
    float mediaAluno = 0.0;
    int notasLidas = 0;
    int quantProvas;

     /* aqui falta ler a quantidade de provas */

     while (notasLidas < quantProvas) {
          /* Aqui falta algo. Olhe no algoritmo. */
     }

    mediaAluno = mediaAluno / quantProvas;
    printf("A media do aluno foi %f\n", mediaAluno);
    if (mediaAluno >= 5.0) {
        printf("Aprovado\n");
    }
    else {
         printf("Reprovado");
    }
    return 0;
}
```