

LINGUAGEM C – Notas de Aula I

Estrutura básica de um programa em C

```
main()  
{  
    /* programa */  
}
```

Obs: Pode-se usar um tipo ou “void” antes do main(), como falaremos mais adiante.

Constantes

Pode-se declarar uma constante com “const”

Exemplos:

```
const x=1;  
const s="abc";
```

Variáveis

Tipos básicos em C

char	8 bits	variações: signed char / unsigned char
int	32 bits	variações: short ou int (short int / long int) signed ou unsigned (signed int / unsigned int)
float	32 bits	
double	64 bits	eventualmente pode haver um long double

(os tamanhos em bits podem variar de acordo com o ambiente usado)

Declaração

tipo nome-da-variável;
exemplo: int x;

Atribuição

com sinal de igualdade (=)
exemplo: x=15;

Declaração e Atribuição

declara e inicializa na mesma sentença.
exemplo: int x=15;

Comandos de Entrada/Saída

Usar uma biblioteca para comandos de entrada/saída (no nosso caso é stdio.h):

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

Leitura de Dados

```
scanf("string de controle", lista de argumentos);
```

na <string de controle>:

%c - leitura de caractere

%d ou %i - leitura de números inteiros

%f - leitura de números reais

%s - leitura de vários caracteres

%ld – inteiros longos

%lf – double longo

Eventualmente, pode-se formatar a saída, como em %8.2f que significa que a saída de um número real ocupará 8 casas no total, mostrando 2 decimais.

A lista de argumentos deve conter exatamente o mesmo número de argumentos quantos forem os códigos de formatação na <string de controle>. Cada variável a ser lida pelo scanf deverá ser precedida pelo caracter & (para strings não - mas o turbo C aceita).

Saída de Dados

```
printf("string de controle", lista de argumentos);
```

Os caracteres a serem utilizados pelo printf() em sua <string de controle> serão os mesmos de scanf(). Necessariamente você precisará ter tantos argumentos quantos forem os comandos de formatação na "string de controle".

Exemplo: Alô, mundo!

```
main()  
{  
    printf("Alô, mundo!");  
}
```

Usando o Turbo C

Teclas úteis:

Compilar – F9

Compilar e Rodar - CTRL F9

Alternar para tela de resultado - ALT F5

Voltar para edição – ALT E

Interromper um programa em loop – CTRL C

Gravar um programa – F2

Carregar um programa salvo – F3

Sair do Turbo C – ALT X

Obs: o editor do Turbo C é muito ruim. Uma alternativa para programas maiores seria usar o Bloco de Notas (notepad), escrever o programa e salvar (no formato nome.c – o nome com até 8 caracteres) no diretório do Turbo C e depois abrir (F3) esse programa no Turbo C.

Exemplo: Programa para ler e mostrar uma idade

```
main()  
{  
    int idade;  
    printf("Digite sua Idade: ");  
    scanf("%d",&idade);  
    printf("Sua idade e' %d anos. \n", idade);  
}
```

Caracteres de escape nas strings de controle:

"\n" para pular linha (o mais usado)
outros: \a Caracter de alerta (bell) \b Retrocesso \f Alimentação de formulário \r Retorno de carro \t
Tabulação horizontal \v Tabulação vertical \\ Contra-barras \? Ponto de interrogação \' Apóstrofo \" Aspas
\ooo Número octal \xhh Número hexadecimal

Exercícios

- Fazer um programa em C para realizar a soma de dois inteiros pedidos ao usuário
- Fazer um programa em C para realizar a média de três inteiros pedidos ao usuário

Operadores

Matemáticos: + - * / % ++ --
Relacionais: < > <= >= != (diferente) == (igual)
Lógicos: && (e) || (ou) ! (não)

Comandos de decisão

```
if (condição(ões))
{
    comando(s);
}
else
{
    comando(s);
}
```

```
switch (variável ou expressão com resultado ordinal)
{
    case valor1: comando(s);
        break;
    case valor2: comando(s);
        break;
    case valor3: comando(s);
        break;
    ...
    default: comando(s);
        break;
}
```

Exemplos básicos para treino:

- dizer se um número é par ou ímpar
- quadrado ou cubo de um número inteiro

Exercício

Escreva um programa que peça três inteiros, correspondentes a dia, mês e ano. Peça os números até conseguir valores que estejam na faixa correta (dias entre 1 e 31, mês entre 1 e 12 e ano entre 1500 e 2500). Verifique se o mês e o número de dias batem (incluindo verificação de anos bissextos).

PS: Um ano é bissexto se for divisível por 4 e não for divisível por 100, exceto para os anos divisíveis por 400, que também são bissextos.

Solução

```
main()
{
    int d,m,a,ok;
    printf("\nDigite uma data (DD/MM/AAAA): ");
    scanf("%d/%d/%d",&d,&m,&a);
    ok=1; /* "flag" usada para indicar que a data está correta */
    if ( (a<1500) || (a>2500) ) { ok=0; }
```

```
else if ( (m<1) || (m>12) ) { ok=0; }
else if ( (d<1) || (d>31) ) { ok=0; }
else if ( ( (m==4) || (m==6) || (m==9) || (m==11) ) && (d==31) ) { ok=0; }
else if ( (m==2) && (d>29) ) { ok=0; }
else if ( (m==2) && !( ( (a%4==0) && (a%100!=0) ) || (a%400==0) ) && (d>28) )
    { ok=0; }
if (ok==0) { printf("\nData Invalida!"); }
else { printf("Data OK!"); }
getch();
}
```