

1) Introdução à Linguagem C (Deitel 2011, cap 2)

Estrutura de um programa em C

Um programa em C é composto por módulos chamados de funções. Uma função contém declarações e instruções que especificam as operações a serem executadas. A função principal chamada main() estabelece o ponto de partida de execução do programa.

```
#include <stdio.h>

/*Definição de constantes e funções */

int main(void) {

    printf("Meu primeiro programa em C.\n");

    return 0;

}
```

Código 1: Primeiro programa em Linguagem C.

Sentença

Cada instrução em C é chamada de sentença. Sentenças simples são terminadas com um ponto e vírgula. Sentenças compostas são blocos delimitados por chaves. O corpo da função main() é um exemplo de sentença composta.

Variáveis

Cada variável possui um identificador (nome) e tipo associado. O nome da variável pode conter letras, underscore e dígitos, não podendo ser iniciada por dígito. Os tipos de dados primários são:

| Tipo de Dados | Bits | Faixa de Valores |
|---------------|------|---------------------------|
| char | 8 | -128 a 127 |
| int | 16 | -32768 a 32767 |
| float | 32 | 7 dígitos significativos |
| double | 64 | 15 dígitos significativos |

Obs.: letra MAIÚSCULA ≠ minúscula

Atribuição

O operador de atribuição (=) permite atribuir um valor a uma variável:

```
char letra;    /* Declaração */  
letra = 'a';   /* Atribuição */
```

```
int num1;  
num1 = 10;
```

Variáveis também podem ser inicializadas enquanto são definidas:

```
float num2 = 123.7;
```

Operadores

Aritméticos: +, -, *, /, %

Relacionais: <, >, <=, >=, ==, !=

Lógicos: ! (negação), && (E lógico), || (OU lógico)

Obs.: Cuidado para não confundir o operador de atribuição (=) com o operador de igualdade (==).

Entrada e Saída

Para utilização das funções de entrada e saída, deve-se incluir a stdio.h:

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

A função printf() pode ser utilizada para imprimir mensagens e valores em uma variedade de formatos. Exemplo:

```
printf("num1 = %d e num2 = %f\n", num1, num2);
```

Usa-se %d para imprimir valores inteiros, %f para valores reais e %c para caracter. Esses símbolos são chamados de especificadores.

A função scanf() pode ser usada para ler valores digitados no teclado. Estes valores são lidos de acordo com os especificadores, que são indicados pelo programador como parâmetros do scanf(). Exemplo:

```
printf("Entre sua idade: ");  
scanf("%d", &idade);  
printf("Voce tem %d anos\n", idade);
```

Mais de um valor pode ser lido pelo mesmo scanf():

```
scanf("%d %d %d", &dia, &mes, &ano);
```

Obs.: Deve-se colocar o & precedendo nomes de variáveis para leitura de tipos primários.

Exemplo: Calcular o perímetro e a área de uma circunferência de raio fornecido pelo usuário.

```
#include <stdio.h> /* funções de entrada-saída */
#include <math.h> /* funções matemáticas */

#define PI 3.14159 /* constante */

int main(void) {

    int raio;
    float perimetro, area;

    printf("Entre com o valor do raio:");
    scanf("%d", &raio);

    perimetro = 2 * PI * raio;
    area = PI * pow(raio, 2);

    printf("O perimetro da circunferencia eh %.2f\n", perimetro);
    printf("A area eh %.2f\n", area);

    return 0;
}
```

Código 2: Cálculo do perímetro e área de uma circunferência.

2) Estruturas Condicionais (Deitel 2011, cap 3)

Estrutura IF e IF-ELSE

Exemplo: Calcular a divisão entre dois números inteiros.

```
#include <stdio.h>

int main(void) {
    int a, b;
    float divisao;

    printf("Entre com dois numeros inteiros: ");
    scanf("%d %d", &a, &b);

    if (b != 0) {
        divisao = a / b;
        printf("Resultado = %f\n", divisao);
    }
    else {
        printf("Divisao por zero.\n");
    }

    return 0;
}
```

Código 3: Exemplo de utilização do comando IF-ELSE.

Construção ELSE-IF

A estrutura IF e IF-ELSE podem ser aninhadas, ou seja, colocadas uma dentro da outra. Por exemplo:

```
if (nota >= 90) {
    printf("Conceito A\n")
}
else {
    if (nota >= 80) {
        printf("Conceito B\n")
    }
    else {
        if (nota >= 70) {
            printf("Conceito C\n")
        }
        else {
            printf("Conceito D\n")
        }
    }
}
```

A construção **ELSE-IF** permite simplificar a decisão entre diversas alternativas.

Exemplo: Imprimir o conceito correspondente a nota informada pelo usuário.

```
#include <stdio.h>

int main(void) {

    float nota;

    printf("Entre com a nota: ");
    scanf("%f", &nota);

    if (nota >= 90) {
        printf("Conceito A\n");
    }
    else if (nota >= 80) {
        printf("Conceito B\n");
    }
    else if (nota >= 70) {
        printf("Conceito C\n");
    }
    else {
        printf("Conceito D\n");
    }

    return 0;
}
```

Código 4: Exemplo de construção ELSE-IF.

3) Exercícios

- Implementar e testar os códigos 2, 3 e 4.
- Pesquisar sobre a estrutura de seleção múltipla SWITCH-CASE.

4) Atividade Avaliativa 1

Faça um programa para calcular expressões matemáticas simples. O programa deve ler a seguinte entrada de dados:

<operando 1> <operador> <operando 2>

Os operandos podem ser números reais ou inteiros. O operador consiste em um caractere, o qual representa uma das operações disponíveis: soma (+), subtração (-), divisão (/), multiplicação (*) e resto da divisão (%). Por fim, o programa deve imprimir o resultado da operação.

Exemplo de execução:

4 * 2

8

Obs.: A entrada do usuário está representada em negrito, enquanto a saída do programa está representada em itálico.

Utilize a estrutura condicional SWITCH-CASE ou a construção ELSE-IF. Para a operação de divisão, esta deve ser realizada apenas se o segundo operando for diferente de zero. Lembre-se que na programação, um caractere deve sempre ser indicado dentro de aspas simples (até mesmo ao realizar comparações).