

Aula 5 - Mais C e repetição

Vamos nesta aula mostrar mais detalhes e a forma geral dos comandos C vistos até agora.

O comando de atribuição

A forma geral do comando de atribuição é:

```
<variável> = <expressão aritmética>
```

A expressão aritmética em C segue praticamente as mesmas regras que uma expressão aritmética em matemática. A prioridade com que as operações são executadas é a usual (* e / e depois + e -). Quando há a mesma prioridade, o cálculo é feito da esquerda para a direita. Os parêntesis alteram a prioridade e podemos ter qualquer quantidade de abre e fecha parêntesis.

Exemplos de comandos de atribuição com expressões aritméticas (por enquanto só com valores inteiros):

```
x = (a+b) / (a-b);  
y = a/2 + 5;  
z = b - 3*(a-c);  
x2 = x*x;
```

O nome de uma variável começa por uma letra (maiúscula ou minúscula) e os demais caracteres podem ser letras ou números e também alguns caracteres especiais (_ e \$). Letras maiúsculas e minúsculas são consideradas diferentes.

O comando scanf

A forma geral é:

```
scanf("<formato>", &<v1>, &<v2>, ... , &<vn>);
```

<v1>, <v2>, ... , <vn> são variáveis.

Até o momento vimos apenas variáveis inteiras (**int**), portanto apenas formato inteiro (**%d**). Existem vários tipos de formato para entrada de dados que veremos no decorrer do curso.

Exemplos – supondo as seguintes variáveis:

```
int x, y, z;
```

```
scanf("%d", &x); - lê um valor e o colocar em x.
```

```
scanf("%d%d%d", &x, &y, &z); - lê 3 valores e coloca respectivamente em x, y e z.
```

O comando printf

A forma geral é:

```
printf("<formato>", <e1>, <e2>, ... , <en>);
```

<e1>, <e2>, ... , <en> são expressões aritméticas.

Existem vários tipos de formato. Por enquanto veremos apenas o formato %d (decimal ou inteiro), cadeias de caracteres e pulo de linha.

Exemplos: supondo $x = 32$, $y = -55$ e $z = 234432$, veja o que será impresso em cada um dos comandos abaixo:

```
printf("Valor de x:%10d", x);
```

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

Valor de x: 32

(note que o valor de x é impresso com 10 casas com brancos à esquerda)

Quando aparece o %d sem especificar a quantidade de casas, o valor é mostrado com a quantidade de casas necessárias.

É sempre melhor colocar alguns brancos separando os valores impressos na mesma linha.

Outra forma é especificar uma quantidade de casas maior do que o tamanho do valor. Assim, os brancos à esquerda separam os valores.

```
printf("x = %5d , y = %5d , x+y = %5d", x, y, x+y);
```

x = 32 , y = -55 , x+y = -23

(note que o valor de x, y e x+y são impressos com 5 casas com brancos à esquerda)

```
printf("x = %5d\ny = %5d\nx+y = %5d", x, y, x+y);
```

**x = 32
y = -55
x+y = -23**

(\n pula para a próxima linha)

Comparações de valores

As comparações de valores podem usar os seguintes operadores:

```
== (igual)  
!= (diferente)  
> (maior)  
< (menor)  
>= (maior ou igual)  
<= (menor ou igual)
```

Exemplos:

```
if (x >= y) ...  
if (a != 2) ...  
if (x > -35) ...
```

O comando if

O formato geral do comando **if** por enquanto é:

```
if (<comparação>) {c1; c2; ... ; cn;}  
else {d1; d2; ... ; dn;}
```

- c1 a cn e d1 a dn são comandos.
- Quando só existe um comando (c1 ou d1), não precisa das chaves.
- A parte do **else** é opcional.

Repetição de Comandos

Suponha o seguinte problema:

Dada uma seqüência de números, calcular a soma destes números.

Se soubéssemos quantos são os números seria fácil:

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

```
soma ← 0;

leia x;
soma ← soma + x;

leia x;
soma ← soma + x;

... (repete-se isso, o número de vezes necessárias)

imprima soma;
```

E se não soubermos a quantidade de números?

Não saberíamos quantas vezes repetir.

Outro problema é que mesmo que soubéssemos a quantidade de números essa forma de resolver seria inconveniente, pois se fossem muitos números o programa ficaria muito extenso.

Além disso, teríamos um programa diferente para cada quantidade de dados.

O ideal seria um só programa para qualquer quantidade de dados.

Vamos modificar um pouco o enunciado do problema:

Dado N e uma seqüência de N números, calcular a soma destes números.

Mesmo com essa modificação não sabemos ainda como repetir N vezes um conjunto de comandos.

Precisamos de alguma forma, contar a quantidade de vezes que um conjunto de comandos é executado e parar quando atingisse esse valor.

Como contar?

Considere a seguinte construção:

```
contador ← 1
Enquanto contador <= N {
    c1; c2; ... ; cn
    contador ← contador + 1;
}
```

Nessa construção, o conjunto de comando `c1; c2; ... ; cn` é repetido N vezes.

Note que a cada repetição o contador `cont` é incrementado de 1.

Em C, existe o comando **while**, justamente para este tipo de construção.

O comando while

O comando `while` do C, tem exatamente a função descrita acima.

Vamos então resolver o problema proposto acima.

P13) Dado $N > 0$ e uma seqüência de N números calcular a soma destes números.

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    int N, soma, x, cont;
    printf("entre com o valor de N:");
    scanf("%d", &N);

    contador = 1; soma = 0;
    while (contador <= N) {
```

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

```
        printf("entre com mais um valor:");
        scanf("%d", &x);
        soma = soma + x;
        contador = contador + 1;
    }
    printf("\nvalor da soma:%10d", soma);
    system("PAUSE");
    return 0;
}
```

O **contador** garante a repetição de **N** vezes. Assim podemos também usar:

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    int N, soma, x, contador;
    printf("entre com o valor de N:");
    scanf("%d", &N);

    contador = 0; soma = 0;
    while (contador < N) {
        printf("entre com mais um valor:");
        scanf("%d", &x);
        soma = soma + x;
        contador = contador + 1;
    }
    printf("\nvalor da soma:%10d", soma);
    system("PAUSE");
    return 0;
}
```

Comentários dentro do programa

Um programa é uma codificação. Podemos colocar comentários no meio da codificação, para tornar mais claro o que estamos querendo fazer. O comentário é uma frase qualquer dentro do programa que não é considerada pelo compilador.

Comentários podem ser colocados dentro do programa precedidos pelos caracteres **/*** e seguidos por ***/**.

Exemplos:

```
/* Entre com N.
   Em seguida entre com N números.
   Calcule a soma dos N números */
```

Veja agora os comentários no programa abaixo.

```
#include <stdio.h>
#include <stdlib.h>
/* Dados N > 0 e uma sequencia de N elementos,
   Calcula a soma dos elementos da seqüência */
int main () {
    /* variáveis usadas no programa */
    int N, soma, x, cont;

    /* leitura da quantidade de elementos */
    printf("entre com o valor de N:");
    scanf("%d", &N);

    cont = 1; /* inicia o contador */
    soma = 0; /* inicia a soma */
```

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

```
/* repete N vezes */
while (cont <= N) {
    printf("entre com mais um valor:");
    scanf("%d", &x);
    soma = soma + x;
    cont = cont + 1;
}
/* imprime o resultado */
printf("\nvalor da soma:%10d", soma);
system("PAUSE");
return 0;
}
```

Consistência dos dados de entrada

No programa anterior, afirmamos no enunciado que $N > 0$.

O que acontece se o digitador entrar com um dado $N \leq 0$?

Em geral é importante que o programa faça consistência nos dados de entrada, para garantir que o digitador ou o usuário do programa está usando o mesmo de forma adequada.

Sem a consistência, o programa pode terminar de forma anormal e não dar informações ao usuário sobre o ocorrido.

Como exemplo, considere a seguinte variação do programa acima:

P14) Dado $N > 0$ e uma seqüência de N números, calcular a média dos elementos desta seqüência.

A solução deste problema é muito parecida com a do anterior. Basta no final dividir a soma dos elementos por N e imprimir esse valor. Ocorre que se o valor de N digitado for 0, o programa terminará de forma anormal porque houve uma tentativa de divisão por 0. É o que se chama de “erro na execução do programa”.

Vamos então resolver esse problema com a consistência dos dados de entrada.

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    /* variáveis usadas no programa */
    int N, soma, x, cont;

    /* leitura da quantidade de elementos */
    printf("entre com o valor de N:");
    scanf("%d", &N);

    /* faz a consistência do valor lido */
    if (N <= 0) {
        /* imprime mensagem e termina o programa */
        printf("deveria ser maior que 0");
        system("PAUSE");
        return -1; /* para diferenciar do retorno normal */
    }

    cont = 1; /* inicia o contador */
    soma = 0; /* inicia a soma */
    /* repete N vezes */
    while (cont <= N) {
        printf("entre com mais um valor:");
        scanf("%d", &x);
        soma = soma + x;
        cont = cont + 1;
    }
    /* imprime o resultado */
}
```

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

```
    printf("\nvalor da media:%10d", soma/N);
    system("PAUSE");
    return 0;
}
```

Nesta solução, se **N** não for maior que zero, terminamos o programa (“**return -1**”).

Melhor seria repetirmos o pedido de entrada até que seja digitado um valor maior que zero para **N**.

Veja agora outra solução com esta modificação:

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    /* variáveis usadas no programa */
    int N, soma, x, cont;

    /* leitura da quantidade de elementos */
    printf("entre com o valor de N:");
    scanf("%d", &N);

    /* faz a consistência do valor lido */
    while (N <= 0) {
        /* imprime mensagem e termina o programa */
        printf("deveria ser maior que 0");
        /* nova leitura da quantidade de elementos */
        printf("entre novamente com o valor de N:");
        scanf("%d", &N);
    }

    cont = 1; /* inicia o contador */
    soma = 0; /* inicia a soma */
    /* repete N vezes */
    while (cont <= N) {
        printf("entre com mais um valor:");
        scanf("%d", &x);
        soma = soma + x;
        cont = cont + 1;
    }
    /* imprime o resultado */
    printf("\nvalor da media:%10d", soma/N);
    system("PAUSE");
    return 0;
}
```

P15) Idem calculando a soma dos positivos e a soma dos negativos.

P15a) Idem calculando a média dos positivos, dos negativos e quantos são nulos.

P15b) Idem verificando se a seqüência de números está em ordem crescente, ou seja cada elemento é maior ou igual ao anterior.

Mais repetição de comandos – outra forma de controlar a quantidade de dados de entrada

Nos exemplos anteriores, controlamos a quantidade de dados de entrada fornecendo como primeiro dado de entrada a quantidade de elementos da seqüência. Isso sempre é possível quando sabemos a priori quantos dados vamos usar. E quando não sabemos? A solução acima pressupõe que o usuário do programa conte com antecedência quando dados irá utilizar.

Uma forma mais simples de fazer a mesma coisa é entrar com um valor especial indicando que os dados acabaram.

Considere a seguinte formulação para o P14.

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

P14a) Dada uma seqüência de números terminada por zero, calcular a média dos elementos da seqüência.

Observe que agora, o último dado a ser digitado é o número zero e ele não faz parte da seqüência. É apenas o marcador de fim. É claro que também a seqüência não pode ter um elemento igual a zero, ou seja, o marcador é um valor especial diferente de todos os demais.

Vamos à solução.

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    /* variáveis usadas no programa */
    int soma, x, cont;

    cont = 0; /* inicia o contador */

    /* le o primeiro elemento */
    printf("entre com o primeiro elemento:");
    scanf("%d", &x);
    soma = x; /* atribui o primeiro elemento a soma */

    /* leitura dos demais elementos */
    while (x != 0) {
        printf("entre com o proximo elemento:");
        scanf("%d", &x);
        soma = soma + x; /* incrementa soma */
        cont = cont + 1; /* incrementa contador */
    }

    /* imprime o resultado */
    printf("\nvalor da media:%10d", soma/cont);
    system("PAUSE");
    return 0;
}
```

Veja que o programa ficou até um pouco menor.

Algumas observações sobre a solução acima:

- 1) Não foi necessário iniciar **soma** com zero, pois o primeiro elemento é que inicia o valor de **soma**.
- 2) O contador **cont** agora começa com zero, pois deve conter exatamente a quantidade de elementos da seqüência, ou seja, a quantidade de dados menos um, pois o zero final não faz parte.
- 3) O zero que indica o final da seqüência também é somado. Felizmente sem alterar o resultado.
- 4) Se a seqüência tiver zero elementos, ou seja, o primeiro número digitado for zero, o programa vai dar erro pois haverá uma divisão por zero. O mesmo ocorre nas soluções anteriores quando N é zero.

Na solução abaixo, a soma é incrementada do valor só depois de verificado que o elemento lido não é zero.

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    /* variáveis usadas no programa */
    int soma, x, cont;

    cont = 0; /* inicia o contador */
    soma = 0; /* inicia soma */

    /* le o primeiro elemento */
    printf("entre com o primeiro elemento:");
    scanf("%d", &x);
```

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11

```
/* soma e leitura do proximo elemento */
while (x != 0) {
    soma = soma + x; /* incrementa soma */
    cont = cont + 1; /* incrementa contador */
    printf("entre com o proximo elemento:");
    scanf("%d", &x);
}

/* imprime o resultado - verificando se a seqüência é nula */
if (cont>0) printf("\nvalor da media:%10d", soma/cont);
else printf("\nsequencia com zero elementos");

system("PAUSE");
return 0;
}
```

Outra forma de resolver é usar um pequeno truque para entrar na repetição, sem precisar ler o primeiro elemento fora da repetição. Veja abaixo.

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    /* variáveis usadas no programa */
    int soma, x, cont;

    cont = 0; /* inicia o contador */
    soma = 0; /* inicia a soma */

    /* Inicia x com um valor diferente de zero para entrar na repeticao.
       Esse valor não será considerado na soma. */
    x = -1;

    /* leitura e soma dos elementos */
    while (x != 0) {
        printf("entre com o proximo elemento:");
        scanf("%d", &x);
        /* soma e conta apenas quando x não for zero */
        if (x != 0) {
            soma = soma + x; /* incrementa soma */
            cont = cont + 1; /* incrementa contador */
        }
    }
    /* imprime o resultado - verificando se a seqüência é nula */
    if (cont>0) printf("\nvalor da media:%10d", soma/cont);
    else printf("\nsequencia com zero elementos");
    system("PAUSE");
    return 0;
}
```

Refaça agora os seguintes exercícios:

P15c) Dada uma seqüência de números (positivos e/ou negativos) terminada por zero, calcular a soma dos positivos e a soma dos negativos.

P15d) Idem verificando se a seqüência de números está em ordem crescente, ou seja cada elemento é maior ou igual ao anterior.

Aula 5 - Mais C e repetição

MAC 110 – Marcilio – Revisado 07Ago11