

## Aula 6 - Mais exemplos de repetição

Vejam agora mais alguns exemplos de problemas envolvendo seqüência de números.

P16) Dado  $N > 0$  e uma seqüência de  $N$  números, determinar o maior elemento da seqüência.

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

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

    /* consistência de N */
    /* ... */

    /* leitura do primeiro elemento */
    printf("entre com um elemento:");
    scanf("%d", &x);
    /* inicia Maior com o primeiro elemento */
    Maior = x;

    cont = 1; /* inicia o contador */
    /* repete N-1 vezes */
    while (cont <= N-1) {
        printf("entre com mais um valor:");
        scanf("%d", &x);
        /* se for maior então troca */
        if (x > Maior) Maior = x;
        cont = cont + 1;
    }
    /* imprime o maior */
    printf("\no maior é :%10d", Maior);
    system("PAUSE");
    return 0;
}
```

P17) Idem calculando o menor

P18) idem o maior, o menor e a média

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

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

    /* consistência de N */
    /* ... */
}
```

```
/* leitura do primeiro elemento */
printf("entre com um elemento:");
scanf("%d", &x);
/* inicia Maior, Menor e Soma com o primeiro elemento */
Maior = x;
Menor = x;
Soma = x;
/* Poderia ser também: Maior=Menor=Soma=x; (atribuição múltipla) */

cont = 1; /* inicia o contador */
/* repete n-1 vezes */
while (cont <= N-1) {
    printf("entre com mais um valor:");
    scanf("%d", &x);
    /* se for maior que Maior então troca */
    if (x > Maior) Maior = x;
    /* se for menor que Menor então troca */
    if (x < Menor) Menor = x;
    /* soma o elemento lido */
    Soma = Soma + x;
    cont = cont + 1;
}
/* imprime o maior, o menor e a media */
printf("\nmaior:%10d\nmenor:%10d\nmédia:%10d", Maior, Menor, Soma/N);
system("PAUSE");
return 0;
}
```

Nos problemas anteriores a quantidade de elementos da sequência N era um dado.  
Vamos agora resolver os mesmos problemas usando um marcador de fim de sequência.

P16a) Dado  $N > 0$  e uma sequência de N números terminada por zero, determinar o maior elemento da sequência.

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    /* variáveis usadas no programa */
    int Maior, x;
    /* leitura do primeiro elemento */
    printf("Entre com um elemento:");
    scanf("%d", &x);
    /* inicia Maior com o primeiro elemento */
    Maior = x;

    /* repete até encontrar elemento igual a zero */
    while (x != 0) {
        /* se for maior então troca */
        if (x > Maior) Maior = x;
        printf("entre com mais um valor:");
        scanf("%d", &x);
    }
    /* imprime o maior */
    printf("\nMaior elemento:%10d", Maior);
    system("PAUSE");
    return 0;
}
```

P17a) Idem calculando o menor

P18a) Idem o maior, o menor e a média

### Operações com inteiros

Estamos usando até agora apenas números inteiros.

O resultado das operações é sempre inteiro, mesmo no caso da divisão, onde o resultado é sempre arredondado para baixo.

Supondo:

```
int i=7, j=3, k; /* podemos atribuir valor à variável na declaração */
```

Qual o valor atribuído a k nos comandos abaixo:

```
k = i/j;          /* 2 e não 2,333... */
k = i/j*j;       /* 6 e não 7 */
k = (i+1)/j;     /* 2 e não 2,666... */
k = (i-2)/j;     /* 1 e não 1,666... */
k = i/(j-1);     /* 3 e não 3,5 */
k = i - i/j*j;   /* 1 e não 0 - o resto da divisão de i por j */
```

Supondo agora:

```
int i=-7, j=3, k;
```

Qual o valor atribuído a k nos comandos abaixo:

```
k = i/j;          /* -2 e não -2,333... */
k = i/j*j;       /* -6 e não -7 */
k = i - i/j*j;   /* -1 e não 0 - o resto da divisão de i por j */
```

### A operação % - resto da divisão

Já vimos nos exemplos acima que o resto da divisão de i por j pode ser calculado pela expressão:

$i - i/j*j$

Para facilitar, existe em C o operador % (resto da divisão).

A expressão que dá o resto da divisão de i por j fica:

$i \% j$

Supondo  $i=7$  e  $j=3$ , qual o valor atribuído a k nos seguintes comandos:

```
k = i \% j;      /* 1 */
k = j \% i;      /* 3 */
k = i * j \% j; /* 0 */
```

E se  $i=-7$  e  $j=3$ , qual o valor atribuído a k:

```
k = i \% j;      /* -1 */
k = j \% i;      /* 3 */
k = i * j \% j; /* 0 */
```

Vamos agora a alguns exemplos que mexem com propriedades dos números.

P19) Dado  $N > 0$  imprimir todos os divisores de  $N$  maiores que 1 e menores que  $N$ .

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

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

    /* consistência de N */
    /* ... */
    div = 2; /* primeiro divisor */
    /* repete para div de 2 até N/2 (último candidato a divisor) */
    while (div <= N/2) {
        if (N % div == 0) printf("\n%5d é um divisor de %5d", div, N);
        div = div + 1;
    }
    system("PAUSE");
    return 0;
}
```

P20) Dado  $N > 1$ , verificar se  $N$  é primo.

Lembrando – um número é primo quando só é divisível por 1 e por ele mesmo.

A primeira solução para esse problema pode ser baseada no anterior. Basta contar o número de divisores de  $N$  e no final, verificar se  $N$  tem 0 ou mais divisores.

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

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

    /* consistência de N */
    /* ... */
    div = 2; /* primeiro divisor */
    ContaDiv = 0; /* inicia o contador de divisores */
    /* repete para div de 2 até N/2 (último candidato a divisor) */
    while (div <= N/2) {
        if (N % div == 0) ContaDiv = ContaDiv + 1; /* mais um divisor */
        div = div + 1;
    }
    /* verifica se N tem divisores */
    if (ContaDiv > 0) printf("%10d - nao primo", N);
    else printf("%10d - primo", N);
    system("PAUSE");
    return 0;
}
```

```
}
```

Você já deve ter percebido que não é necessário contar os divisores. Ao encontrar o primeiro divisor, o programa pode parar e imprimir "**não primo**".

Vamos então à solução.

Modificando um pouco mais, podemos iniciar a variável **div** na declaração.

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    /* variáveis usadas no programa */
    int N, div=2; /* iniciar div na declaração */

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

    /* consistência de N */
    /* ... */

    /* repete para div de 2 até N/2 (último candidato a divisor) */
    while (div <= N/2) {
        if (N % div == 0) {
            printf("%10d - não primo", N);
            system("PAUSE");
            return 0; /* sai do programa */
        }
        div = div + 1;
    }
    /* se veio até aqui, N não tem divisores */
    printf("%10d - primo", N);
    system("PAUSE");
    return 0;
}
```

Existem outras melhorias que podem ser feitas neste programa:

- Não precisa testar os divisores até  $N/2$ . Basta testar até a raiz quadrada de  $N$ . É um resultado da álgebra. Como não vimos ainda, a função raiz quadrada, resolva o problema usando inteiros. Basta testar:  
**while (div\*div <= N) ...**
- Não precisa testar para todos os possíveis divisores de 1 em 1. Basta testar para 2 e depois só os ímpares.
- Na verdade basta testar somente para todos os primos entre 2 e raiz quadrada de  $N$ . Ocorre que não temos estes primos. Vamos deixar esta solução para mais tarde.