



PCI- Estrutura de Repetição II

Profa. Mercedes Gonzales
Márquez

Estrutura de Repetição

Exemplo 1: Imprimir os n primeiros números inteiros positivos.

```
int i, n;  
scanf("%d",&n);  
for (i=1; i<=n;i++)  
    printf("\n %d",i);
```

Exemplo 2: Faça um algoritmo que determine os quadrados de um conjunto de números inteiros positivos.

```
int num;  
scanf("%d",&num);  
for (; num>0;){  
    printf("%d\n",num*num);  
    scanf("%d",&num);  
}
```

Estrutura de Repetição

Exemplo 3: Imprimir os divisores de um número natural n

Para isso, passamos por todos os números naturais i até o número n e verificamos se cada um deles é um divisor de n . Se o resto da divisão de n por i for zero, então estamos diante de um divisor de n .

```
int i, n;  
printf("Entre com um inteiro positivo: ");  
scanf("%d", &n);  
for (i = 1; i <= n; i++)  
    if (n % i == 0) /* Se o resto for zero, então i é divisor */  
        printf("%d\n", i);
```

Estrutura de Repetição

Sobre o exemplo 3 algumas observações se fazem necessárias:

(A) Se um número inteiro X possui um divisor Y menor que sua raiz quadrada, o quociente da divisão de X por Y será maior que a raiz quadrada de X e será, também, um divisor de X .

(B) Se um número inteiro X possui um divisor Y igual a sua raiz quadrada, o quociente da divisão de X por Y será o próprio divisor Y .

Estrutura de Repetição

A implementação anterior é uma das implementações mais ineficientes possíveis, pois vários dos candidatos da lista $[2; 3; \dots; n - 1]$ podem ser definitivamente descartados. Por exemplo, nenhum número maior que $n/2$ pode ser divisor de n , pois isso significaria que n também tem um divisor entre 1 e 2. Ainda mais, se só precisamos verificar se n é primo (e não achar todos os seus divisores), basta verificar os possíveis divisores $d \leq \text{raiz}(n)$ (pois, se d for realmente divisor, $n/d \geq \text{raiz}(n)$ também será divisor).

Mas, no momento, para o propósito atual, vamos nos contentar com a maneira ineficiente.

Estrutura de Repetição

Exemplo 4: Faça um programa que dado um inteiro positivo N , imprima se ele é um número primo ou não.

Para testar se um número n é primo devemos verificar se ele tem um divisor não-trivial, isto é, um que não seja igual a 1 ou n . Assim, verificamos entre todos os candidatos (a princípio, $[2; 3; : : : ; n - 1]$, mas essa lista ainda pode ser bem refinada, como comentado no exercício anterior) se há algum divisor de n . Se não acharmos nenhum, é porque n é primo.

Estrutura de Repetição

Exemplo 4: Faça um programa que dado um inteiro positivo n , imprima se ele é um número primo ou não.

```
int n, i, /* candidato a divisor */
divisores=0; /* contador de divisores */
printf("Entre com um inteiro positivo: ");
scanf("%d", &n);
for (i = 2; i < n; i++) /* testa todos os candidatos a divisor */
    if (n % i == 0)
        divisores++;
if (divisores == 0) /* não achamos nenhum divisor dentre os
                    possíveis candidatos , então é primo */
    printf("O numero %d é primo\n", n);
else
    printf ("O numero %d nao é primo\n", n);
```

Estrutura de Repetição – Comandos break e continue

Comando break

O comando break faz com que a execução de um laço de repetição seja finalizada, passando a execução para o próximo comando após o laço. Exemplo:

```
int i;  
for (i = 1; i <= 5; i++) {  
    if (i > 3)  
        break;  
    printf("%d\t", i);  
}  
printf("Fim\n");
```

O que será impresso?

1 2 3 Fim

Estrutura de Repetição

Comandos break e continue

Exemplo 1: Faça um programa que dado um inteiro positivo n , imprima se ele é um número primo ou não.

```
int n, i;
printf("\nDigite um número natural: ");
scanf("%u", &n);
for(i=2; i<=n-1; i++)
    if( n%i == 0 )
        break;
if( i==n )
    printf("\nO número é primo ");
else
    printf("\nO número não é primo");
```

7 % 2 == 1

7 % 3 == 1

7 % 4 == 3

7 % 5 == 2

7 % 6 == 1

∴ 7 é um número primo

Só podemos chegar a essa conclusão depois de ter verificado todas as possibilidades!

9 % 2 == 1

9 % 3 == 0

∴ 9 não é um número primo

Chegamos a essa conclusão assim que encontramos o primeiro divisor entre 1 e n !

Estrutura de Repetição – Comandos break e continue

Comando continue

O comando continue faz com que a execução da iteração corrente do laço de repetição seja finalizada, passando a execução para a próxima iteração do laço. Exemplo:

```
int i;  
for (i = 1; i <= 5; i++) {  
    if (i == 3)  
        continue;  
    printf("%d\t", i);  
}  
printf("Fim\n");
```

O que será impresso?

1 2 4 5 Fim

Combinação de Estruturas: Estrutura Condicional dentro de Repetições

- **Exemplo 1:** Escrever um programa que receba dois números inteiros positivos, e determine o produto dos mesmos, utilizando o seguinte método de multiplicação:
 - dividir, sucessivamente, o primeiro número por 2, até que se obtenha 1 como quociente;
 - paralelamente, dobrar, sucessivamente, o segundo número;
 - somar os números da segunda coluna que tenham um número ímpar na primeira coluna. O total obtido é o produto procurado.

Exemplo:

$$\begin{array}{rcl} & & 12 \times 6 \\ 12 & 6 & \\ 6 & 12 & \\ 3 & 24 \rightarrow & 24 \\ 1 & 48 \rightarrow & +48 \\ & & \hline & & 72 \end{array}$$

Combinação de Estruturas: Estrutura Condicional dentro de Repetições

```
int a,b,orig_a, orig_b,pro=0;
printf ("Informe dois inteiros: ");
scanf ("%d %d", &a,&b);
orig_a=a;
orig_b=b;
while (a>=1) {
    if (a%2!=0)
        pro+=b; /* Acumulador*/
    a/=2; /* atualização de a*/
    b*=2; /* atualização de b*/
}
printf ("O produto de %d e %d e' %d",orig_a,orig_b,pro);
```

Combinação de Estruturas- Estrutura Repetição dentro de Condicional

Agora veremos o contrário da combinação anterior, isto é, um comando de repetição no escopo do comando de desvio condicional.

Exemplo 2: Imprimir o Maximo Divisor Comum (MDC) entre dois números dados.

Combinação de Estruturas- Estrutura Repetição dentro de Condicional

Nós aprendemos na escola a calcular o MDC da seguinte forma. Vejo o exemplo: Calcular o MDC entre 72 e 135.

72	135	3
24	45	3
8	15	MDC= 3x3=9

Combinação de Estruturas- Estrutura Repetição dentro de Condicional

O conceito matemático de MDC entre dois números dados x e y envolve a fatoração de cada número como um produto de fatores primos, escolhendo-se os fatores primos em comum com a potência mínima.

$$72 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 3$$

$$135 = 3 \cdot 3 \cdot 3 \cdot 5$$

Conclui-se assim, que o MDC entre 72 e 135 é $3 \cdot 3$, pois o 3 é o único fator primo em comum e a menor potência comum é 2.

Outro exemplo: 12 e 8

$$12 = 2 \cdot 2 \cdot 3$$

$$8 = 2 \cdot 2 \cdot 2$$

$$\text{MDC} = 2 \cdot 2 = 4$$

Combinação de Estruturas- Estrutura Repetição dentro de Condicional

O máximo divisor comum entre dois números naturais não nulos e distintos é igual ao máximo divisor entre o menor e o resto da divisão do maior pelo menor.

Em símbolos, se x e y são números naturais tais que $x > y > 0$, e se

x	y
r	q

então $\boxed{mdc(x, y) = mdc(y, r)}$.

Exemplo: $mdc(12, 18) == mdc(18, 12) == mdc(12, 6) == mdc(6, 0) == 6$.

Baseado nesse resultado temos que:

A ideia é dividir o maior número pelo menor, e depois fazer divisões sucessivas do último divisor pelo último resto, até obtermos uma divisão exata.

Exemplo:

1320		35		35		25		25		10		10		5
25		37		10		1		5		2		0		2

Exemplo: $mdc(1320, 35) == mdc(35, 25) == mdc(25, 10) == mdc(10, 5) == mdc(5, 0) = 5$

Estrutura Repetição dentro de Condicional

```
int main(){
    int x, y, resto;
    printf ("Informe dois numeros inteiros para calcular o MDC: ");
    scanf ("%d %d",&x,&y);
    if (x!=0 && y!=0){
        resto = x%y;
        while (resto!=0){
            x =y;
            y =resto ;
            resto =x%y;
        }
        printf ("mdc = %d", y) ;
    }
    else
        printf (" Nao funciona para entradas nulas.")
}
```

Combinação de Estruturas- Estrutura Repetição Aninhadas

Vamos ver problemas para os quais os algoritmos exigem o aninhamento de repetições.

Exemplo 3: Imprimir as tabuadas do 1 ao n .

Estrutura Repetição Aninhadas

```
#include <stdio.h>
int main(){
    int i ,j,n;
    printf ("Informe um numero inteiro: ");
    scanf("%d",&n);
    for (i=1;i <= n;i++){
        printf ("Tabuada do %d\n", i);
        for (j=1;j <= 10;j++)
            printf ( "%d x %d = %d\n", i,j, i*j ); //comando mais interno
    }
}
```

Combinação de Estruturas- Estrutura Repetição Aninhadas

Exemplo 4: Imprimir o valor do fatorial de todos os números entre 1 e n, sendo n fornecido pelo usuário.

O fatorial de n é assim definido:

$$n = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$$

No exercício 5 vimos o programa para o fatorial de um único número n e agora veremos o programa para o fatorial de um conjunto de números.

Estrutura Repetição Aninhadas

```
#include <stdio.h>
int main(){
    int i , n, fat=1;
    printf ("Informe um numero n: ");
    scanf ("%d", &n);
    fat =1; /*inicializacao do acumulador */
    for (i=n; i>= 1;i--)
        fat = fat * i;
    printf ("O fatorial de %d eh %d",n,fat);
}
```

Estrutura Repetição Aninhadas

```
/* Fatorial de um conjunto de inteiros*/
```

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

```
int main(){
```

```
    int i , n, fat, quant;
```

```
    printf ("Informe a quantidade de inteiros dos quais iremos calcular o  
fatorial: ");
```

```
    scanf ("%d", &quant);
```

```
    for (n=1; n<=quant; n++) {
```

```
        fat =1; /*inicializacao do acumulador */
```

```
        for (i=n; i>= 1;i--)
```

```
            fat = fat * i;
```

```
        printf ("O fatorial de %d eh %d",n,fat);
```

```
    }
```

Combinação de Estruturas- Estrutura Repetição Aninhadas

O programa funciona, mas é extremamente ineficiente e repleto de cálculos redundantes. Perceba a seguinte propriedade de fatorial:

$$n! = n \times (n-1)!$$

Exemplo:

$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$ e $5! = 5 \times 4 \times 3 \times 2 \times 1$ então

$$6! = 6 \times 5!$$

Ou seja quando você calcular o fatorial de n você pode aproveitar o cálculo do fatorial do número anterior, isto é, de $n-1$.

Assim sendo, temos uma versão mais eficiente do programa anterior.

Combinação de Estruturas- Estrutura Repetição Aninhadas

```
#include <stdio.h>
int main(){
    int i , n, fat=1, quant;
    printf ("Informe a quantidade de inteiros dos quais iremos calcular o
fatorial: ");
    scanf ("%d", &quant);
    for (n=1;n<=quant;n++) {
        fat = fat * n ;
        printf ("O fatorial de %d eh %d",n,fat);
    }
}
```