

Estruturas de Repetiçãõ

Lista de Exercícios - 04

Linguagem e Técnicas de Programação



Professor: Edwar Saliba Júnior

Estruturas de Repetiçãõ

O que são e para que servem?

São comandos que são utilizados na programação quando se deseja repetir, determinada parte do código, mais de uma vez. Ou seja, ao invés de escrevermos o mesmo código duas ou mais vezes, utilizamos uma estrutura de repetição. Exemplo:

Queremos imprimir os números inteiros de 1 a 10 no vídeo do computador.

a) Sem estrutura de repetição

```
início  
|  
|   escreva "1"  
|   escreva "2"  
|   escreva "3"  
|   escreva "4"  
|   escreva "5"  
|   escreva "6"  
|   escreva "7"  
|   escreva "8"  
|   escreva "9"  
|   escreva "10"  
fim
```

```
int main(){  
    printf("1");  
    printf("2");  
    printf("3");  
    printf("4");  
    printf("5");  
    printf("6");  
    printf("7");  
    printf("8");  
    printf("9");  
    printf("10");  
  
    return 0;  
}
```

b) Com estrutura de repetição PARA

```
início  
|  
declare Cont : inteiro  
  
|  
para Cont ← 1 até 10 passo 1 faça  
|  
    escreva Cont  
fim para  
  
fim
```

```
int main(){  
    int cont;  
  
    for(cont = 1; cont <= 10; cont++){  
        printf("%d", cont);  
    }  
  
    return 0;  
}
```

c) Com estrutura de repetição ENQUANTO

```
início  
|  
declare Cont : inteiro  
  
|  
Cont ← 1  
  
|  
enquanto (Cont <= 10) faça  
|  
    escreva Cont  
|  
    Cont ← Cont + 1  
fim enquanto  
  
fim
```

```
int main(){  
    int cont;  
  
    cont = 1;  
    while(cont <= 10){  
        printf("%d", cont);  
        cont = cont + 1;  
    }  
  
    return 0;  
}
```

d) Com estrutura de repetição REPITA¹

```
início
|
| declare Cont : inteiro
|
| Cont ← 1
|
| repita
| | escreva Cont
| | Cont ← Cont + 1
| enquanto (Cont <= 10)
|
fim
```

```
int main(){
    int cont = 1;

    do{
        printf("%d", cont);
        cont++;
    }while(cont <= 10);

    return 0;
}
```

Observação: Para este caso especificamente, não há muito problema em escrevermos o comando “Escreva” seguido de um número, dez vezes em nosso algoritmo. Por outro lado, imagine como ficaria nosso algoritmo, sem estrutura de repetição, se quiséssemos imprimir os números inteiros de 1 a 10.000.

Nota: As estruturas de repetição também são conhecidas por: *LOOPS* ou *LAÇOS*.

Para, Enquanto e Repita

Vimos acima que existem três estruturas de repetição diferentes: a estrutura PARA, a estrutura ENQUANTO e a estrutura REPITA. Até aí tudo bem, mas, quando utilizarmos cada uma delas?

Vejam!

¹ A estrutura apresentada é exclusiva para a Linguagem de Programação C e suas derivadas. No caso da Linguagem de Programação Pascal e suas derivadas, a estrutura passa a ser: REPITA ... ATÉ (Condição). Há diferenças na forma de teste da condição, nestas estruturas.

1. Estrutura PARA

Deverá ser utilizada quando se sabe previamente o número de repetições que deverão ser executadas. Exemplo:

Imprima todos os números pares no intervalo de 1 a 30.

Para este problema, já foi determinado o número de vezes que o *loop* será executado, ou seja, 30 vezes.

Resolução do problema:

```
início
|
declare Cont : inteiro
|
para Cont ← 1 até 30 passo 1 faça
|   se (Cont mod 2 = 0) então /* Testa se Cont possui valor par. */
|       escreva Cont
|   fim para
|
fim
```

```
int main(){
    int cont;

    for(cont = 1; cont <= 30; cont++){
        if(cont % 2 == 0){
            printf("%d", cont);
        }
    }

    return 0;
}
```

2. Estrutura REPITA

Deverá ser utilizada quando o *loop* tem que ser executado no mínimo uma vez e, a execução do *loop* mais de uma vez estará sujeita à condição imposta no final. Exemplo:

Imprima o somatório de todos os números inteiros no intervalo de 0 (zero) a N. Onde N deve ser um número inteiro maior ou igual a zero e será escolhido pelo usuário.

Para este problema, podemos considerar que, no mínimo uma vez o *loop* deverá ser feito, pois, o menor número que o usuário poderá digitar é o 0 (zero).

Resolução do problema:

```
início
|
| declare N, Total, Cont : inteiro
|
| Total ← 0
|
| escreva "Digite um número inteiro maior ou igual a zero: "
| leia N
|
| Cont ← 0
|
| repita
| | Total ← Total + Cont
| | Cont ← Cont + 1
| enquanto (Cont <= N)
|
| escreva "O Somatório do intervalo de 0 a N é: ", Total
|
fim
```

```
int main(){
    int N, total = 0, cont = 0;

    printf("Digite um número inteiro maior ou igual a zero: ");
    scanf("%d", &N);

    do{
        total = total + cont;
        cont++;
    }while(cont <= N);

    printf("O somatório do intervalo de 0 a N é: %d", total);

    return 0;
}
```

3. Estrutura ENQUANTO

Deverá ser utilizada quando, antes de se executar o *loop*, for necessário testar uma condição.

Imprima o resultado da operação X^Y (leia-se: X elevado a Y). Onde X é a base e o primeiro número que o usuário digitará, e Y é o expoente ou potência e será o segundo número a ser digitado. Ambos inteiros.

Para este problema deveremos fazer o teste da condição antes de entrarmos na estrutura de repetição.

Resolução do problema:

```
início
|
| declare base, exp, total : inteiro
|
| escreva "Digite o valor da base: "
| leia base
| escreva "Digite o valor do expoente: "
| leia exp
|
| total ← 1
|
| enquanto (exp > 0) faça
| | total ← total * base
| | exp ← exp - 1
| fim enquanto
|
| escreva "O resultado da potenciação é: ", total
fim
```

```
int main(){
    int base, exp, total = 1;

    printf("Digite o valor da base: ");
    scanf("%d",&base);
    printf("Digite o valor do expoente: ");
    scanf("%d",&exp);

    while(exp > 0){
        total = total * base;
        exp--;
    }

    printf("O resultado da potenciação é: %d", total);

    return 0;
}
```

Exercícios

- 1) Elabore um programa que solicite que o usuário entre com 10 números inteiros. Some estes números, calcule a média aritmética e imprima seu valor.
- 2) Elabore um programa que solicite que o usuário entre com um número inteiro qualquer. Imprima uma linha de asteriscos (*) com a quantidade digitada pelo usuário. Ex.:

- 3) Elabore um programa em que o usuário entre com um número inteiro qualquer. Seu *software* deverá somar os 20 números subsequentes ao que foi digitado pelo usuário e imprimir o resultado.
- 4) Faça um programa que imprima a multiplicação, divisão, subtração e adição de 1 até 9, para um número digitado pelo usuário do *software*.
- 5) Faça um programa que solicite que o usuário entre com o seu sexo (M ou F). Enquanto o valor lido não for o correto (M ou F), exibir mensagem de erro e solicitar que o usuário entre com um novo valor. Quando o valor que o usuário digitar estiver correto, imprimir na tela o valor do sexo por extenso. (Ex.: Se digitado "F", imprimir "Feminino")
- 6) Faça um programa que leia valores inteiros digitados pelo usuário até que seja digitado o valor 0 (zero). Quando o valor zero for digitado, deverá ser mostrado na tela a quantidade de valores que foram digitados e também a soma destes (até antes do valor zero).
- 7) Elabore um programa que some todos os números inteiros de 1 até 528. Apresente o resultado desta soma.
- 8) Elabore um programa que solicite que o usuário entre com um valor inteiro inicial e um valor inteiro final. Some todos os números deste intervalo, calcule a média aritmética e apresente na tela uma lista contendo todos os valores do intervalo digitado que estão acima da média calculada.
- 9) Escreva um *software* que calcule o valor da série: $S = 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + \dots + 1/50$.
- 10) Escreva um programa que calcule o valor da série: $S = 1/1 + 3/2 + 5/3 + 7/4 + 9/5 + \dots + 99/50$.
- 11) Para o exercício 11, calcule a média aritmética da série e apresente uma lista contendo todos os valores da série que estão acima da média.
- 12) Elabore um programa que leia um número qualquer digitado pelo usuário e calcule seu Fatorial. (Exemplo: $5! = 5 \times 4 \times 3 \times 2 \times 1$)
- 13) Elabore um programa em que o usuário entre com um número inteiro qualquer, e o *software* imprima os 20 números subsequentes ao que foi digitado pelo usuário.
- 14) Elabore um programa que solicite que o usuário entre com 300 números quaisquer. Ao final apresente separadamente:
 - a. A soma dos 100 primeiros números digitados;
 - b. A soma do 101º número até o 200º;
 - c. A soma do 201º número até o 300º.
- 15) Elabore um programa que apresente os números pares existentes no intervalo fechado [A, B]. Sendo que A e B serão números inteiros escolhidos pelo usuário. Um número é par quando este satisfaz a seguinte condição: $(\text{NÚMERO} \% 2 == 0)$

- 16) Elabore um programa que solicite que o usuário entre com 100 números quaisquer. Ao final apresente separadamente:
- A soma dos números pares que existirem entre o 1º número digitado até 50º;
 - A soma dos números ímpares que existirem entre o 51º número digitado até o 100º.
- 17) Escreva um programa que solicite que o usuário entre com valores inteiros quaisquer. Ao final imprima a quantidade de números digitados, o somatório dos valores digitados, e a média aritmética do somatório.
- 18) Elabore um programa para fazer cálculo de potenciação. Ou seja, x^y . (Exemplo: $3^4 = 3 \times 3 \times 3 \times 3 = 81$). Seu *software* deverá solicitar que o usuário entre com o valor da base (x) e do expoente (y). Para resolver o problema não deverá ser usada a função `pow` da biblioteca `math.h`, utilize estrutura de repetição. Para este problema serão considerados apenas números inteiros positivos.
- 19) Elabore um programa que apresente todos os números primos no intervalo de 1 a 50. Um número é considerado Primo quando ele puder ser dividido exclusivamente por 1 e por ele próprio.

Mais informações e exercícios poderão ser encontrados em:

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. de. **Fundamentos da Programação de Computadores** : Algoritmos, Pascal e C/C++, São Paulo: Pearson, 2002.

- Páginas: 79 a 124.

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. de. **Fundamentos da Programação de Computadores** : Algoritmos, Pascal, C/C++ e Java, 2ª. Ed., São Paulo: Pearson, 2007.

- Páginas: 93 a 144.

Exercícios de Depuração Usando Estruturas de Repetição

1) Apresente o que será impresso na tela do computador pelos programas a seguir:

a) void main(){

```
int J, I, X;
J = 100;
X = 3;
J = J + 40;
I = 5 * X - 4;
while(X >= 5){
    J = J - 15;
    X++;
    I = I + X - J;
}
printf("J = %d, I = %d, X = %d", J, I, X);
}
```

b) void main(){

```
int J, I, X;
J = 100;
X = 3;
J = J + 40;
I = 5 - X * 4;
do{
    J = J - 15;
    X = X + 1;
    I = I + X - J;
}while(X >= 5);
printf("J = %d, I = %d, X = %d", J, I, X);
}
```

c) void main(){

```
int J, I, X;
J = 100;
X = 3;
J = J + 40;
I = 5 - X * 4;
while(X <= 5){
    J = J - 15;
    X += 1;
    I = I + X - J;
}
printf("J = %d, I = %d, X = %d", J, I, X);
}
```

d) void main(){

```
int M, N, Y;
M = 10;
Y = 1;
for(N = 1; N <= 6; N += 2){
    M = M - 8;
    Y = Y * 3 + N;
}
printf("\nM = %d, Y = %d, N = %d", M, Y, N);
}
```

e) void main(){

```
int P, Q;
float VALOR;
P = 5;
Q = P - 8;
VALOR = 18;
do{
    VALOR = VALOR + (VALOR * P + 100);
    P = P + 2;
    Q = Q + 1;
}while(Q < 0);
printf("Valor = %f", VALOR);
}
```

f) void main(){

```
int CONT;
float VALOR;
char RESP;
CONT = 0;
VALOR = 0;
RESP = 's';
while(RESP == 's'){
    VALOR = VALOR + 139;
    CONT++;
    if(CONT > 3){
        RESP = 'n';
    }
}
printf("Valor = %f", VALOR);
}
```

g) void main(){

```
int N;
float SOMA;
SOMA = 0;
for(N = 1; N <= 5; N++){
    SOMA = SOMA + 1 / N;
}
printf("Soma = %f", SOMA);
}
```

h) void main(){

```
int N;
N = 0;
while(N < 5){
    if(N == 0){
        printf("Esse número não existe: 1/0");
    } else {
        printf("\n %f", 1 / N);
    }
    N++;
}
}
```