

Estruturas de Repetição

Lista de Exercícios - 04



Programação de Computadores I

Professor: Edwar Saliba Júnior

Estruturas de Repetição

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

```
1  #include<stdio.h>
2
3  int main() {
4      printf("\n 1");
5      printf("\n 2");
6      printf("\n 3");
7      printf("\n 4");
8      printf("\n 5");
9      printf("\n 6");
10     printf("\n 7");
11     printf("\n 8");
12     printf("\n 9");
13     printf("\n 10");
14
15     return 0;
16 }
17
```

b) Com estrutura de repetição FOR (PARA)

```
1  #include<stdio.h>
2
3  int main() {
4      int i;
5
6      for(i = 1; i <= 10; i++){
7          printf("\n %d", i);
8      }
9
10     return 0;
11 }
12
```

c) Com estrutura de repetição WHILE (ENQUANTO)

```
1  #include<stdio.h>
2
3  int main(){
4      int i = 1;
5
6      while(i <= 10){
7          printf("\n %d", i);
8          i++;
9      }
10
11     return 0;
12 }
13
```

d) Com estrutura de repetição DO...WHILE (REPITA ENQUANTO¹)

```
1  #include<stdio.h>
2
3  int main(){
4      int i = 1;
5
6      do{
7          printf("\n %d", i);
8          i++;
9      }while(i <= 10);
10
11     return 0;
12 }
13
```

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

Nota: As estruturas de repetição também são conhecidas por: *LOOP* ou *LAÇO*.

Para, Enquanto e Repita Enquanto

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?

Vejamos!

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 diferença 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:

```
1  #include<stdio.h>
2
3  int main(){
4      int cont;
5
6      for(cont = 1; cont <= 30; cont++){
7          if(cont % 2 == 0){
8              printf("\n %d", cont);
9          }
10     }
11
12     return 0;
13 }
14
```

2. Estrutura REPITA ENQUANTO

Deverá ser utilizada quando os comandos dentro do *loop* têm que ser executados no mínimo uma vez. E a execução do *loop* mais de uma vez, estará sujeita à condição imposta no final da estrutura. 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 lupe deverá ser feito, pois, o menor número que o usuário poderá digitar é o 0 (zero).

Resolução do problema:

```
1  #include<stdio.h>
2
3  int main(){
4      int N, total = 0, cont = 0;
5
6      printf("Digite um número inteiro maior ou igual a zero: ");
7      scanf("%d", &N);
8
9      do{
10         total = total + cont;
11         cont++;
12     }while(cont <= N);
13
14     printf("O somatório do intervalo de 0 a N é: %d", total);
15
16     return 0;
17 }
18
```

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 (restringiremos nosso cálculo de potência aos números inteiros, para diminuir a complexidade do programa, isto no intuito de facilitar o aprendizado).

Para este problema deveremos fazer o teste da condição antes de entrarmos no *loop*.

Resolução do problema:

```
1  #include<stdio.h>
2  |
3  int main(){
4      int x, y, total = 1;
5
6      printf("Digite o valor da base: ");
7      scanf("%d", &x);
8      printf("Digite o valor do expoente: ");
9      scanf("%d", &y);
10
11     while(y > 0){
12         total = total * x;
13         y--;
14     }
15     printf("\n O resultado de x^y é: %d", total);
16
17     return 0;
18 }
19
```

Exercícios

- 1) Elabore um programa que solicite que o usuário entre com 100 números inteiros quaisquer. Imprima a soma dos números digitados.
- 2) Elabore um programa que leia um número qualquer digitado pelo usuário e calcule seu Fatorial. (Exemplo: $5! = 5 \times 4 \times 3 \times 2 \times 1$)
- 3) 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.
- 4) Elabore um programa que solicite que o usuário entre com dois números (inicial e final). Ao final o programa deverá apresentar o valor total da soma de todos os números do intervalo digitado pelo usuário.
- 5) 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.
- 6) Elabore um programa que apresente os números pares maiores que 10 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: $(\text{NÚMERO} \% 2 = 0)$
- 7) Elabore um programa que solicite que o usuário entre com 100 números quaisquer. Ao final apresente separadamente:
 - a. A soma dos números pares que existirem entre o 1º número digitado até 50º;
 - b. A soma dos números ímpares que existirem entre o 51º número digitado até o 100º.
- 8) 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.
- 9) Elabore um programa para fazer cálculo de potenciação. Ou seja, x^y . (Exemplo: $3^4 = 3 \times 3 \times 3 \times 3$). Seu programa deverá solicitar que o usuário entre com o valor da base (x) e do expoente (y) e apresentar o resultado do cálculo sem utilizar os operadores ** ou ^. Para resolver o problema utilize estrutura de repetição.
- 10) Escreva um programa que calcule a média da seguinte sequência numérica a seguir: $1/2 + 1/3 + 1/4 + 1/5 + 1/6 + \dots + 1/50$. Feito isto, o programa deverá apresentar uma lista contendo todos os números da sequência que estão acima da média calculada.
- 11) 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.