



LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA

EXERCÍCIOS COM REPETIÇÃO E REPETIÇÃO ANINHADA

Prof. Dr. Daniel Caetano

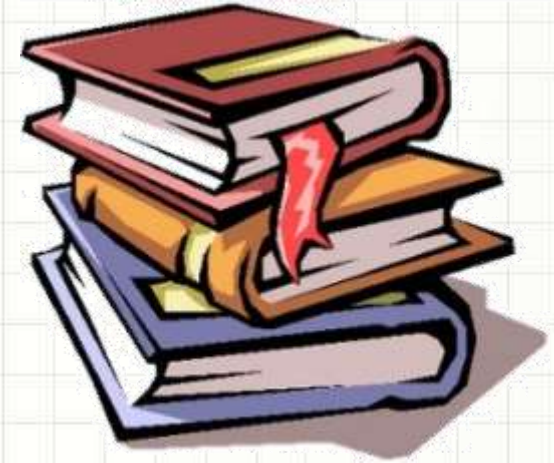
2012 - 1

Objetivos

- Treinar o desenvolvimento de algoritmos que envolvam repetição
- Compreender algoritmos com repetição aninhada
- **PARA CASA**
 - Lista de Exercícios 2 está **ONLINE!**



Material de Estudo



Material

Acesso ao Material

Tutorial

<http://www.caetano.eng.br/aulas/lpe/>
(Aula 16)

Apresentação

<http://www.caetano.eng.br/aulas/lpe/>
(Aula 16) – PARCIAL / COMPLETO

Material Didático

Fundamentos da Programação de Computadores –
Parte 2 – Páginas 93 a 144.



EXERCÍCIOS COM REPETIÇÕES

EXERCÍCIO

1) Faça um programa que receba dois inteiros N1 e N2 e calcule a soma de todos os números entre eles (inclusive N1 e N2).

Exemplo:

Se $N1 = 10$ e $N2 = 16$, o programa deve calcular

$$10 + 11 + 12 + 13 + 14 + 15 + 16$$

EXERCÍCIO

```
1) #include <stdio>
#include <iostream>
using namespace std;
int main(void) {
    int N1, N2, SOMA;
    cout << "Digite o primeiro número: ";
    cin >> N1;
    cout << "Digite o segundo número: ";
    cin >> N2;
    SOMA = 0;
    do {
        SOMA = SOMA + N1;
        N1 = N1 + 1;
    } while (N1 <= N2);
    cout << "Resultado: " << SOMA << endl;
    getch();
}
```

EXERCÍCIO

2) Faça um programa que receba o salário atual do funcionário (SAL) a taxa de aumento anual (TAXA) e o número de anos (ANOS) e calcule o salário do funcionário depois que esses anos se passarem.

EXERCÍCIO

2)

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void) {

    float SAL, TAXA;

    int ANOS;

    cout << "Digite o salário atual: ";
    cin >> SAL;
    cout << "Digite a taxa de reajuste anual: ";
    cin >> TAXA;
    cout << "Digite o número de anos: ";
    cin >> ANOS;

    while (ANOS > 0) {
        SAL = SAL + (SAL * TAXA);
        ANOS = ANOS - 1;
    }

    cout << "Salário final: " << SAL << endl;

    getch();
}
```


EXERCÍCIO

2)

```
#include <stdio>

float SAL, TAXA;

int ANOS;

cout << "Digite o salário atual: ";
cin >> SAL;
cout << "Digite a taxa de reajuste anual: ";
cin >> TAXA;
cout << "Digite o número de anos: ";
cin >> ANOS;

while (ANOS > 0) {
    SAL = SAL + (SAL * TAXA);
    ANOS = ANOS - 1;
}

cout << "Salário final: " << SAL << endl;
}
```

EXERCÍCIO

3) Faça um programa que leia o número de cidades N e, para cada uma delas, pergunte o número de nascimentos (NASC) no último ano. O programa deve responder o número de nascimentos da cidade onde ocorreu mais nascimentos e também a média de nascimentos por cidade.

EXERCÍCIO

3)

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void) {

    int N, CIDADES, NASC;
    int MAX, TOTAL;
    float MEDIA;

    cout << "Digite o número de cidades: ";
    cin >> CIDADES;

    TOTAL = 0;
    MAX = 0;
    for (N = CIDADES; N > 0; N = N - 1) {
        cout << "Quantos nascimentos na cidade " << N << "? ";
        cin >> NASC;

        TOTAL = TOTAL + NASC;
        if (NASC > MAX) MAX = NASC;
    }

    MEDIA = (float)TOTAL/(float)CIDADES;

    cout << "Máximo: " << MAX << endl;
    cout << "Média: " << MEDIA << endl;

    getchar();
}
```

EXERCÍCIO

```
int N, CIDADES, NASC;  
int MAX, TOTAL;  
float MEDIA;  
  
cout << "Digite o número de cidades: ";  
cin >> CIDADES;  
  
TOTAL = 0;  
MAX = 0;  
for (N = CIDADES; N > 0; N = N - 1) {  
    cout << "Quantos nascimentos na cidade " << N << "? ";  
    cin >> NASC;  
  
    TOTAL = TOTAL + NASC;  
    if (NASC > MAX) MAX = NASC;  
}  
  
MEDIA = (float)TOTAL/(float)CIDADES;  
  
cout << "Máximo: " << MAX << endl;  
cout << "Média: " << MEDIA << endl;  
  
    getch();  
}
```



REPETIÇÕES ANINHADAS

Repetições Aninhadas

- Repetições aninhadas são repetições...
 - Com outras repetições dentro

```
for (C=1; C<=10; C=C+1) {  
    for (D=0; D<10; D=D+1) {  
        R = C * D;  
        cout << R << endl;  
    }  
}
```

- O que faz esse código?

EXERCÍCIO

4) Faça um programa que desenhe um quadrado 3x3 usando asteriscos:

```
* * *  
* * *  
* * *
```

- Use repetições aninhadas!

```

/*****
E)  Desenha um Quadrado 3 x 3

```

```

4) *****/

```

```

#include <stdio>
#include <iostream>
using namespace std;

int main(void) {

    int i, j;

    // Imprime cada uma das linhas
    for (i = 0; i < 3; i = i + 1) {

        // Imprime cada uma das colunas
        for (j = 0; j < 3; j = j + 1) {
            cout << "*";
        }

        cout << endl; // Ao fim de cada linha, "pula linha"
    }

    getch();
}

```


EXERCÍCIO

5) Faça um programa que desenhe um retângulo 6x3 usando asteriscos:

```
* * * * * *  
* * * * * *  
* * * * * *
```

- Use repetições aninhadas!

```
/******
```

EX Desenha um Retângulo 6 x 3

5)

```
*****/
```

```
#include <stdio>
#include <iostream>
using namespace std;
```

```
int main(void) {
```

```
    int i, j;
```

```
    // Imprime cada uma das linhas
    for (i = 0; i < 3; i = i + 1) {
```

```
        // Imprime cada uma das colunas
        for (j = 0; j < 6; j = j + 1) {
            cout << "*" << " ";
        }
```

```
        cout << endl; // Ao fim de cada linha, "pula linha"
    }
```

```
    getch();
```

```
}
```

EXERCÍCIO

6) Faça um programa que desenhe um trapézio retângulo de bases 3 e 6, usando asteriscos:

```
* * *  
* * * *  
* * * * *  
* * * * * *
```

- Use repetições aninhadas!

EXERCÍCIO

6)

```
/******  
Desenha um Trapézio de Bases 3 e 6  
***  
****  
*****  
*****  
*****/  
  
#include <stdio>  
#include <iostream>  
using namespace std;  
  
int main(void) {  
  
    // Valor inicial das colunas  
    int icol = 3;  
    int fcol = 6;  
    int i, j;  
  
    // Imprime cada uma das linhas  
    for (i = icol; i <= fcol; i = i + 1) {  
  
        // Imprime cada uma das colunas  
        for (j = 0; j < i ; j = j + 1) {  
            cout << "*";  
        }  
  
        cout << endl;  
    }  
  
    getch();  
}
```

EXERCÍCIO

```
6) // Valor inicial das colunas
int icol = 3;
int fcol = 6;
int i, j;

// Imprime cada uma das linhas
for (i = icol; i <= fcol; i = i + 1) {

    // Imprime cada uma das colunas
    for (j = 0; j < i ; j = j + 1) {
        cout << "*";
    }

    cout << endl;
}
```

EXERCÍCIO

7) Modifique o programa anterior para que ele desenhe um triângulo retângulo de base 4, usando asteriscos:

```
*  
* *  
* * *  
* * * *
```

- Use repetições aninhadas!

EXERCÍCIO

7)

```
/******  
Desenha um Triângulo de Base 4  
***  
****  
*****  
*****  
*****/  
  
#include <stdio>  
#include <iostream>  
using namespace std;  
  
int main(void) {  
  
    // Valor inicial das colunas  
    int icol = 1;  
    int fcol = 4;  
    int i, j;  
  
    // Imprime cada uma das linhas  
    for (i = icol; i <= fcol; i = i + 1) {  
  
        // Imprime cada uma das colunas  
        for (j = 0; j < i ; j = j + 1) {  
            cout << "*";  
        }  
  
        cout << endl;  
    }  
  
    getch();  
}
```

EXERCÍCIO

7)

```
// Valor inicial das colunas
int icol = 1;
int fcol = 4;
int i, j;

// Imprime cada uma das linhas
for (i = icol; i <= fcol; i = i + 1) {

    // Imprime cada uma das colunas
    for (j = 0; j < i ; j = j + 1) {
        cout << "*";
    }

    cout << endl;
}
```


EXERCÍCIO

8) Faça um programa que imprima as tabuadas do 1 ao 10, neste formato:

Tabuada do 1: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

Tabuada do 2: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20.

Tabuada do 3: 3, 6, 9, 12, 15, 18, 21, 24, 27, 30.

Tabuada do 4: 4, 8, 12, 16, 20, 24, 28, 32, 36, 40.

Tabuada do 5: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50.

Tabuada do 6: 6, 12, 18, 24, 30, 36, 42, 48, 54, 60.

Tabuada do 7: 7, 14, 21, 28, 35, 42, 49, 56, 63, 70.

Tabuada do 8: 8, 16, 24, 32, 40, 48, 56, 64, 72, 80.

Tabuada do 9: 9, 18, 27, 36, 45, 54, 63, 72, 81, 90.

Tabuada do 10: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100.

- Use repetições aninhadas!

EXERCÍCIO

8)

```
/******  
    Imprime Tabuada do 1 ao 10  
*****/  
#include <stdio>  
#include <iostream>  
using namespace std;  
  
int main(void) {  
  
    int i, j;  
  
    // Imprime cada uma das linhas (cada tabuada)  
    for (i = 1; i <= 10; i = i + 1) {  
  
        cout << "Tabuada do " << i << ": ";  
  
        // Imprime cada uma das colunas (cada operação)  
        for (j = 1; j <= 10 ; j = j + 1) {  
            cout << (i*j);  
            if (j != 10) {  
                cout << ", ";  
            }  
            else {  
                cout << ".";  
            }  
        }  
  
        cout << endl;  
    }  
  
    getch();  
}
```

EXERCÍCIO

8) `int i, j;`

```
// Imprime cada uma das linhas (cada tabuada)  
for (i = 1; i <= 10; i = i + 1) {
```

```
    cout << "Tabuada do " << i << ": ";
```

```
    // Imprime cada uma das colunas (cada operação)
```

```
    for (j = 1; j <= 10 ; j = j + 1) {
```

```
        cout << (i*j);
```

```
        if (j != 10) {
```

```
            cout << ", ";
```

```
        }
```

```
    else {
```

```
        cout << " .";
```

```
    }
```

```
    }
```

```
    cout << endl;
```

```
}
```



CONCLUSÕES

Resumo

- As estruturas de repetição são de grande utilidade para alguns tipos de cálculo
- As estruturas de repetição podem ser aninhadas para gerar resultados combinados
- É preciso cuidado ao analisar o que uma estrutura de repetição faz

- **TAREFA!**
 - AV2!

Próxima Aula



THE END!



PERGUNTAS?



**BOM DESCANSO
A TODOS!**