

FCT/Unesp – Presidente Prudente
Departamento de Matemática e Computação

ATP I

Comandos de Repetição

do while
for

Prof. Danilo Medeiros Eler
danilo.eler@unesp.br

Comandos de Repetição

- Comandos de repetição permitem que blocos de códigos sejam repetidos diversas vezes
- Geralmente, a repetição é finalizada quando uma determinada condição é satisfeita
 - Por exemplo, repetir enquanto:
 - quantidade < 10
 - opcao $\neq 5$
 - numero $\% 2 == 0$

Comandos de Repetição

- Os comandos de repetição em uma linguagem de programação podem ser utilizados em diferentes situações
- Por isso, são disponibilizados comandos para atender a, pelo menos, três casos:
 - Repetição com Teste no Início
 - Repetição com Teste no Final
 - Repetição Contada

Comando de Repetição *while*

- A sintaxe do comando *while* é a seguinte

```
while (condição){  
    instruções;  
}
```
- A condição deve resultar em verdadeiro ou falso—na linguagem C: em 1 ou 0

Comando de Repetição *do while*

- A sintaxe do comando *do while* é a seguinte

```
do {  
    instruções;  
}while(condição);
```
- A condição deve resultar em verdadeiro ou falso—na linguagem C: em 1 ou 0

Comando de Repetição *do while*

- A sintaxe do comando *do while* é a seguinte

```
do {  
    instruções;  
}while(condição);
```

```
while (condição){  
    instruções;  
}
```

- A condição deve resultar em verdadeiro ou falso—na linguagem C: em 1 ou 0

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

Comando de Repetição *do while*

```
→ int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	0
-------	---

Comando de Repetição *do while*

```
int opcao;  
do{  
→ printf("1 - Exemplo opcao 1\n");  
  printf("2 - Exemplo opcao 2\n");  
  printf("3 - Exemplo opcao 3\n");  
  printf("4 - Exemplo opcao 4\n");  
  printf("5 - Sair\n");  
  scanf("%d",&opcao);  
  //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	0
-------	---

1 – Exemplo opção 1

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	0
-------	---

1 – Exemplo opção 1
2 – Exemplo opção 2

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    → printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	0
-------	---

1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	0
-------	---

1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	0
-------	---

1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	3
-------	---

1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
3

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    → //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	3
-------	---

1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
3
Opção 3

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
→ }while(opcao != 5);
```

MEMÓRIA

opcao	3
-------	---

1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
3
Opção 3

opcao != 5
3 != 5
Verdadeiro

Comando de Repetição *do while*

```
int opcao;  
do{
```

```
→ printf("1 - Exemplo opcao 1\n");  
printf("2 - Exemplo opcao 2\n");  
printf("3 - Exemplo opcao 3\n");  
printf("4 - Exemplo opcao 4\n");  
printf("5 - Sair\n");  
scanf("%d",&opcao);  
//switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	3
-------	---

1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
3
Opção 3
1 – Exemplo opção 1

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	3
-------	---

1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
3
Opção 3
1 – Exemplo opção 1
2 – Exemplo opção 2

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	3
-------	---

2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
3
Opção 3
1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    → printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	3
-------	---

3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
3

Opção 3

1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    → printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	3
-------	---

4 – Exemplo opção 4
5 – Exemplo opção 5
3
Opção 3
1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    → scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	5
-------	---

5 – Exemplo opção 5
3
Opção 3
1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
5

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    → //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	5
-------	---

3
Opção 3
1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
5
Opção 5

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
→ }while(opcao != 5);
```

MEMÓRIA

opcao	5
-------	---

3
Opção 3
1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
5
Opção 5

opcao != 5
5 != 5
Falso

Comando de Repetição *do while*

```
int opcao;  
do{  
    printf("1 - Exemplo opcao 1\n");  
    printf("2 - Exemplo opcao 2\n");  
    printf("3 - Exemplo opcao 3\n");  
    printf("4 - Exemplo opcao 4\n");  
    printf("5 - Sair\n");  
    scanf("%d",&opcao);  
    //switch case para tratar as opções  
}while(opcao != 5);
```

MEMÓRIA

opcao	5
-------	---

3
Opção 3
1 – Exemplo opção 1
2 – Exemplo opção 2
3 – Exemplo opção 3
4 – Exemplo opção 4
5 – Exemplo opção 5
5
Opção 5



Comando de Repetição *for*

- A sintaxe do comando *for* é a seguinte
for(inicialização; condição; incremento){
 instruções;
}
- Exemplo:
for (int i=0; i<5; i++){
 printf("%d ",i);
}
- A condição deve resultar em verdadeiro ou falso—na linguagem C: em 1 ou 0

Comando de Repetição *for*

- A sintaxe do comando *for* é a seguinte
for(inicialização; condição; incremento){
 instruções;
}

- Exemplo:

```
for (int i=0; i<5; i++){  
    printf("%d",i);  
}
```

```
int i=0;  
while (i<5){  
    printf("%d ",i);  
    i++;  
}
```

- A condição deve resultar em verdadeiro ou falso—na linguagem C: em 1 ou 0

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

Comando de Repetição *for*

```
→ int num1, num2;  
   printf("Digite valor do primeiro numero: ");  
   scanf("%d", &num1);  
   printf("Digite valor do segundo numero: ");  
   scanf("%d", &num2);  
   for(int i=num1; i<=num2; i++){  
       printf("%d ",i);  
   }
```

MEMÓRIA

num1	0
num2	0

Comando de Repetição *for*

```
int num1, num2;
```

```
→ printf("Digite valor do primeiro numero: ");
```

```
scanf("%d", &num1);
```

```
printf("Digite valor do segundo numero: ");
```

```
scanf("%d", &num2);
```

```
for(int i=num1; i<=num2; i++){
```

```
    printf("%d ",i);
```

```
}
```

MEMÓRIA

num1	0
num2	0

Digite o valor do primeiro numero:

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
→ scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	0

Digite o valor do primeiro numero: 2

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
→ printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	0

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero:

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
→ scanf("%d", &num2);  
for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
→ for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5
i	2

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
→ for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

i<=num2
2<=5
verdadeiro

MEMÓRIA

num1	2
num2	5
i	2

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5
i	2

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
→ for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5
i	3

i++
2++
3

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
→ for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

i<=num2
3<=5
verdadeiro

MEMÓRIA

num1	2
num2	5
i	3

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5
i	3

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2 3

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
→ for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5
i	4

i++
3++
4

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2 3

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
→ for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

i<=num2
4<=5
verdadeiro

MEMÓRIA

num1	2
num2	5
i	4

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2 3

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
for(int i=num1; i<=num2; i++){  
→ printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5
i	4

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2 3 4

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
→ for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5
i	5

i++
4++
5

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2 3 4

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
→ for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

i<=num2
5<=5
verdadeiro

MEMÓRIA

num1	2
num2	5
i	5

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2 3 4

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
for(int i=num1; i<=num2; i++){  
→ printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5
i	5

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2 3 4 5

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
→ for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5
i	6

i++
5++
6

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2 3 4 5

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
→ for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

i<=num2
6<=5
falso

MEMÓRIA

num1	2
num2	5
i	6

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2 3 4 5

Comando de Repetição *for*

```
int num1, num2;  
printf("Digite valor do primeiro numero: ");  
scanf("%d", &num1);  
printf("Digite valor do segundo numero: ");  
scanf("%d", &num2);  
for(int i=num1; i<=num2; i++){  
    printf("%d ",i);  
}
```

MEMÓRIA

num1	2
num2	5
i	6

Digite o valor do primeiro numero: 2
Digite o valor do segundo numero: 5
2 3 4 5

Exercícios

Fazer os seguintes programas com o comando **do while**

- 1) Ler cinco valores fornecidos pelo usuário e calcular a média dos valores.
- 2) Ler quantos valores serão fornecidos pelo usuário. Em seguida, fazer a leitura desses valores. Por fim, apresentar a soma e a média.
- 3) Ler 5 valores do usuário e apresentar o maior valor, o menor valor, a soma e a média dos valores.
- 4) Ler valores inteiros positivos do usuário até que o número 0 seja fornecido. Ao final, apresentar a quantidade de números pares e ímpares.
- 5) Ler um número e apresentar o seu fatorial. Lembrando que:

$$n! = 1 \times 2 \times 3 \times 4 \times \dots \times n-1 \times n$$

$$0! = 1$$

$$\text{Exemplo: } 5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$$

Exercícios

Fazer os seguintes programas com o comando ***for***

6) Ler cinco valores fornecidos pelo usuário e calcular a média dos valores.

7) Ler quantos valores serão fornecidos pelo usuário. Em seguida, fazer a leitura desses valores. Por fim, apresentar a soma e a média.

8) Ler 5 valores do usuário e apresentar o maior valor, o menor valor, a soma e a média dos valores.

9) Ler valores inteiros positivos do usuário até que o número 0 seja fornecido. Ao final, apresentar a quantidade de números pares e ímpares.

10) Ler um número e apresentar o seu fatorial. Lembrando que:

$$n! = 1 \times 2 \times 3 \times 4 \times \dots \times n-1 \times n$$

$$0! = 1$$

$$\text{Exemplo: } 5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$$

11) Apresentar a tabuada dos número de 0 a 10. Para tanto, utilize dois comandos *for* aninhados (um comando *for* dentro de outro comando *for*)

Bibliografia Básica

1. ASCENCIO, A. F. G.; CAMPOS, E. A. V. C. **Fundamentos da programação de computadores**: algoritmos, pascal e C/C++. Pearson Prentice Hall, 2003. 355p.
2. KERNINGHAN, B. W.; Ritchie, D. M. **C: a Linguagem de Programação padrão ANSI**. Rio de Janeiro: Editora Campus, 1990. 289p.
3. KERNINGHAN, B. W.; Pike, R. **A Prática de Programação**. Rio de Janeiro: Editora Campus, 2000, 280p.
4. LOPES, A.; GARCIA, G. **Introdução à Programação**: 500 exercícios resolvidos. Rio de Janeiro: Editora campus, 2002. 469p.
5. MANZANO, J. A. N. G.; OLIVEIRA, J. F. **Algoritmos**: lógica para desenvolvimento de programação de computadores. 10ª ed. Editora Érica Ltda. 2000, 236p.
6. MIZRAHI, V. V. **Treinamento em Linguagem C – Curso Completo – Módulo 1**. São Paulo: MAKRON Books do Brasil Editora Ltda, 1990. 241p.
7. MEDINA, M.; FERTIG, C. **Algoritmos e programação**: teoria e prática. São Paulo: Novatec Editora. 384p. 2005.
7. PUGA, S.; RISSETTI, G. **Lógica de programação e estruturas de dados**: com aplicações em JAVA. São Paulo: Pearson Education do Brasil, 254p. 2004.
8. SCHILDT H. **C Completo e Total**. 3ª ed. São Paulo: MAKRON Books do Brasil editora Ltda. 1997. 827p.
9. XAVIER, G. F. C. **Lógica de Programação**. São Paulo: Editora SENAC. 1998. 378p.

Bibliografia Complementar

1. BROOKSHEAR, J. G. **Ciência da computação**: uma visão abrangente. 5ª ed., Bookman Editora, 2000. 499p.
2. CORMEN, T.H., Leiserson, C.E., Rivest R.L., Stein, C. **Algoritmos**: teoria e Prática. Rio de janeiro: Editora Campus, 2002. 916p.
3. PLAUGER, P. L. **A Biblioteca Standard C**. Rio de Janeiro: Editora Campus, 1994. 614p.
4. PRATA, S. **C primer plus**, 4ª ed. SAMS Publishing, 2002. 931p.
5. OLIVEIRA, U. **Programando em C**, vol. I – fundamentos. Editora Ciência Moderna, 2008, 743p.