

# Algoritmos

## Estrutura de Repetição

**“para - passo”**

**for**

---

Grupos de Slides No 7.

Prof. SIMÃO

# Estrutura de Repetição “para passo”

---

**for** ( *Variável = ValorIni : Incremento : ValorFin* )

*conjunto de comandos;*

**end**

# Exemplo de Algoritmo – 1A – V1

```
% algoritmo/programa para imprimir os números de 1 a 1000.
```

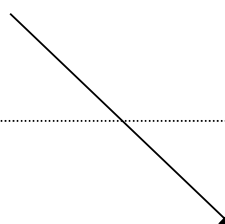
```
clc;
```

```
for ( num = 1 : 1000 )
```

```
    printf ( ' Número %d: \n ', num);
```

```
end
```

```
% fim do algoritmo/programa.
```



Obs. A variável num é inicializada de forma “automática” (implicitamente) e o incremento dela também é “automático”.

# Exemplo de Algoritmo – 1A – V2

```
% algoritmo/programa para imprimir os números de 1 a 1000.
```

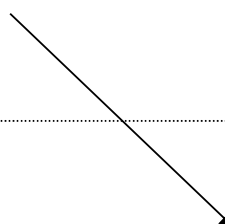
```
clc;
```

```
for ( num = 1 : 1 : 1000 )
```

```
    printf ( 'Número %d: \n', num);
```

```
end
```

```
% fim do algoritmo/programa.
```



Obs. A variável num é inicializada de forma “automática” (implicitamente) e o incremento dela também é “automático” podendo ser explicitado.

# while – for – do until

---

**% Enquanto – while**

```
clc;  
num = 1;  
while ( num <= 1000 )  
    printf ( ' Número %d: \n ', num);  
    num = num + 1;  
end
```

**% Parra passo – for**

```
clc;  
for ( num = 1 : 1 : 1000 )  
    printf ( ' Número %d: \n ', num);  
end
```

**~~% Repita até – do until~~**

```
clc;  
num = 1;  
do  
    printf ( ' Número %d: \n ', num);  
    num = num + 1;  
until ( num > 1000 )
```

# Exemplo de Algoritmo - 2

---

```
% algoritmo/programa para imprimir os números de 1 a 1000.  
  
clc;  
  
soma = 0;  
  
for ( num = 1 : 1 : 1000 )  
    soma = soma + num;  
  
end  
  
printf ( ' O somatório dos números entre 1 e 1000 é: %d. \n ', soma);  
  
% fim do algoritmo/programa.
```

# Explicando o Exemplo

**% algoritmo/programa para imprimir os números de 1 a 1000.**

**clc;**

**soma = 0;**

**for ( num = 1 : 1 : 1000 )**

**soma = soma + num;**

**end**

**printf ( ' O somatório dos números entre 1 e 1000 é: %d. \n ', soma);**

**% fim do algoritmo/programa.**

Passo	Num	Soma
1	2	1
2	3	3
3	4	6
4	5	10
5	6	15
6	7	21
7	8	28
8	9	36
...	...	...

# Exercício 1

---

- **Algoritmo para somar todos os números de uma sequência que começa por um e finaliza em um número dado pelo usuário.**
- **Obs.: Usar a estrutura *for*.**



# Solução Exercício 1

---

```
% algoritmo/programa para imprimir os números de 1 a 1000.
```

```
clc;
```

```
soma = 0;
```

```
printf ( ' Soma de 1 até um número dado. \n ' );
```

```
fim = input( ' Informe um número: ' );
```

```
for ( num = 1 : 1 : fim )
```

```
    soma = soma + num;
```

```
end
```

```
printf ( ' O somatório dos números entre 1 e %d é: %d. \n ', fim, soma );
```

```
% fim do algoritmo/programa.
```

# Exercício 2

---

- Algoritmo para somar os números ímpares entre 5 e 500 (inclusive).

Obs. Utilizar a estrutura *for*.

# Solução Ex.2

**% algoritmo/programa para soma os número impares entre 5 e 500**

**soma = 0;**

**for ( num = 5 : 2 : 499 )**

**soma = soma + num;**

**end**

passo	soma	num
0	0	5
1	5	7
2	12	9
3	21	11
4	32	13
5	45	15
6	60	17

**printf ( ' O somatório dos números ímpares entre 5 e 500 é: %d. \n ' , soma);**

**% Fim do algoritmo/Programa.**

# Exercício 3

---

- **Elaborar um algoritmo para o cálculo da soma, subtração, multiplicação ou divisão de dois números reais fornecidos pelo usuário, segundo sua opção.**
- **O usuário poderá realizar quantas operações desejar enquanto não optar por sair do programa.**

**Obs. Utilizar a estrutura *for*.**

# Solução exercício.

```
% algoritmo/programa para operações elementares sobre dois números
clc;
opcao = 0;
for ( num = 1 : 1 : 1000 )
    printf ( ' Operações elementares sobre dois números. \n ' );
    printf ( ' Digite 1 para soma. \n ' );
    printf ( ' Digite 2 para subtração. \n ' );
    printf ( ' Digite 3 para multiplicação. \n ' );
    printf ( ' Digite 4 para divisão. \n ' );
    printf ( ' Digite 5 para sair. \n ' );
    opcao=input( ' Informe sua opcao: ' );
    printf ( '\n' );
    if ( ( opcao >= 1 ) && ( opcao <= 4 ) )
        num1 = input( ' Digite o primeiro numero: ' );
        num2 = input( ' Digite o segundo numero: ' );
        end

        switch ( opcao )
            case 1
                soma = num1 + num2;        printf ( ' O valor da soma é %f: \n ', soma);
            case 2
                sub = num1 - num2;        printf ( ' O valor da subtração é %f: \n ', sub);
            case 3
                mult = num1 * num2;        printf ( ' O valor da multiplicação é %f: \n ', mult);
            case 4
                if ( num2 != 0 )
                    div = num1 / num2;    printf ( ' O valor da divisão é %f: \n ', div);
                else
                    printf ( ' Divisão por zero impossível! \n ' );
                end
            case 5
                printf ( ' Fim da execução do programa. \n ' ); break;
            otherwise
                printf ( ' Opção inválida. \n ' );

        end
    end
    printf("\n \n");
end
```

# Exercício 4

---

**4.1 - Elaborar um algoritmo para receber as notas de 150 alunos e calcular/apresentar a média das notas.**

**4.2 - Elaborar um algoritmo para receber as 4 notas de cada um dos 150 alunos, calculando/apresentando a média de cada um, bem como a média geral da turma.**

**Obs.: Em ambos, utilizar *para-passo*.**

# Solução para Exercício 4.1

```
% algoritmo/programa para calcular a média de 150 notas.
```

```
clc;
```

```
nota = 0;
```

```
soma = 0;
```

```
quantidade = 150;
```

```
flag = 1;
```

```
for ( cont = 1 : 1 : quantidade )
```

```
    while ( flag == 1 )
```

```
        flag = 0;
```

```
        printf ( ' Informe a %d a nota (entre 0 e 10): ', cont);
```

```
        nota = input( ' ' );
```

```
        if ( ( nota < 0 ) || ( nota > 10 ) )
```

```
            printf ( ' Nota não válida. \n ' );
```

```
            flag = 1;
```

```
        end
```

```
    end
```

```
    soma = soma + nota;
```

```
    flag = 1;
```

```
end
```

```
media = soma / quantidade;
```

```
printf ( ' O valor da média eh: %. \n ' , media);
```

```
% fim do algoritmo/programa.
```

# Exercícios 5.

---

- Algoritmo para permitir ao usuário escolher entre o cálculo do cubo, do quadrado ou da raiz quadrada de um número dado por ele. O usuário também pode escolher como opção 'sair do programa'.
  - Uma primeira versão usando apenas estrutura de decisão *se (if end)* para tratar a opção do usuário.
  - Uma segunda versão usando apenas estrutura de decisão *se senão (if else end)* para tratar a opção do usuário.
  - Uma terceira versão usando apenas estrutura de decisão *escolha caso (switch case)* para tratar a opção do usuário.
  - Para cada uma das versões anteriores usar a estrutura *para-passo (for end)*, comparando com as soluções realizadas anteriormente com a estrutura *enquanto (while end)*.



# Exercícios 6

---

- **6.1 Algoritmo para permitir o cálculo da área ou do perímetro de uma circunferência.**
  - Raio fornecido pelo usuário.
  - Depois de cada cálculo o algoritmo deve permitir ao usuário escolher a mesma ou outra opção.
  - Utilizar a estrutura *escolha-caso (switch case)*.
  - Utilizar uma variável *caracter* para tratar a opção do usuário.
  - O algoritmo só terminará quando o usuário escolher uma opção de término.
  - Utilizar a estrutura *parra-passo (for)*.
- **6.2 Algoritmo para cálculo da área de um quadrado, de um triângulo retângulo ou de uma circunferência dependendo da escolha do usuário.**
  - Parâmetros (e.g. lado ou raio) fornecidos pelo usuário.
  - Utilizar a estrutura *escolha-caso (switch case)*.
  - Utilizar uma variável tipo *caracter* para tratar a opção do usuário.
  - Depois de cada cálculo o algoritmo deve permitir ao usuário escolher a mesma ou outra opção.
  - O algoritmo só terminará quando o usuário escolher uma opção de término.
  - Utilizar a estrutura *para-passo (for)*.

# Exercícios - 7

---

- **7.1 - Algoritmo para o cálculo do quadrado e da raiz quadrada de um número dado pelo usuário.**
- **7.2 Algoritmo para o cálculo do fatorial de um número dado pelo usuário.**
  - **Obs.: Usar a estrutura de repetição *for*.**

# Algoritmo Fatorial

```
% algoritmo/programa para o cálculo do fatorial
clc;
numero=input( ' Informe um número decimal inteiro: ' );

if ( numero > 1 )

    resultado = numero;

    for ( numero = numero : -1 : 2 )
        resultado = resultado * (numero -1);
    end

    printf ( ' O fatorial é: %f \n ', resultado );

else

    if ( numero >= 0 )
        printf ( ' O fatorial é 1. \n ' );
    else
        printf ( ' Número inválido. \n ' );
    end

end

% fim do algoritmo/programa
```

Numero = 5

	resultado	numero
	5	5
5	5*4=20	4
4	20*3=60	3
3	60*2=120	2
2	120*1=120	1

# Algoritmo Fatorial V.2

```
% algoritmo/programa para o cálculo do fatorial
```

```
clc;
```

```
numero=input( ' Informe um número decimal inteiro: ' );
```

```
if ( numero > 1 )
```

```
    resultado = 1;
```

```
    for ( numero = numero : -1 : 1 )
```

```
        resultado = resultado * numero;
```

```
    end
```

```
    printf ( ' O fatorial é %d: \n ', resultado);
```

```
else
```

```
    if ( numero >= 0 )
```

```
        printf ( ' O fatorial eh 1. \n ' );
```

```
    else
```

```
        printf ( ' Número inválido. \n ' );
```

```
    end
```

```
end
```

```
% fim do algoritmo/programa
```

numero = 5

	resultado	numero
	1	5
5	1*5=5	4
4	5*4=20	3
3	20*3=60	2
2	60*2=120	1
1	120*1=120	0

# Exercício

---

- Refazer os todos os exercícios por meio de Diagrama de Atividades da UML, que são fluxogramas modernos por assim dizer.
- Opcionalmente:
  - (Re) Pesquisar sobre Fluxogramas tradicionais.
  - Refazer todos os exercícios anteriores por meio de fluxogramas tradicionais.