

Algoritmos

Estrutura de Repetição

“enquanto”

while

Grupos de Slides No 5.

Prof. SIMÃO

Estrutura de Repetição “enquanto”

% enquanto

while (*condição for verdadeira*)

conjunto de comandos

end

Exemplo de Algoritmo - 1

```
% algoritmo/programa para imprimir os números de 1 a 1000.  
  
clc;  
  
num = 1;  
  
while ( num <= 1000 )  
  
    printf ( 'Número: %i \n', num);  
  
    num = num + 1;  
  
end  
  
% Fim do algoritmo/programa.
```

Exemplo de Algoritmo - 2

```
% Algoritmo/Programa para somar os nuhmeros de 1 a 1000.
```

```
clc;
```

```
soma = 0;
```

```
num = 1;
```

```
while ( num <= 1000 )
```

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

```
    num = num + 1;
```

```
end
```

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

```
% Fim do algoritmo/programa.
```

Explicando o Exemplo

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

clc;

soma = 0;

num = 1;

while (num <= 1000)

soma = soma + num;

num = num + 1;

end

printf ('O somatório dos números entre 1 e 1000 é: %i. \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.

Solução Exercício 1

```
% algoritmo/programa para imprimir os números de 1 a X.  
  
clc;  
  
soma = 0;  
  
num = 1;  
  
printf ( 'Soma de 1 até um número dado. \n');  
  
fim = input( 'Informe um número: ');  
  
while ( num <= fim )  
  
    soma = soma + num;  
  
    num = num + 1;  
  
end  
  
printf ( 'O somatório dos números entre 1 e %f eh: %f. \n', fim, soma );  
  
% fim do algoritmo/programa.
```

Exercício 2

- **2.1 - Algoritmo para decidir se um número é par ou impar.**
- **2.2 - Algoritmo para decidir se cada número entre 1 e 500 é par ou impar.**
- **2.3 - Algoritmo para somar os números pares entre 5 e 500 (inclusive).**

Solução 2.1

```
% algoritmo/Programa para decidir se um número é par ou impar.
```

```
num=input ( 'Digite um numero: ' );
```

```
resto = rem ( num , 2 );
```

```
if ( resto == 0 )
```

```
    printf ( 'É um número par! \n' );
```

```
else
```

```
    printf ( 'É um número impar! \n' );
```

```
end
```

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

Solução 2.2

```
% algoritmo/programa para classificar os número entre 1 e 500  
% como par ou impar.
```

```
num = 1;
```

```
while ( num <= 500 )
```

```
    resto = rem ( num , 2 ) ;
```

```
    if ( resto == 0 )
```

```
        printf ( 'O número %d é par! \n', num);
```

```
    else
```

```
        printf ( 'O número %d é impar! \n', num);
```

```
    end
```

```
    num = num + 1;
```

```
end
```

```
% Fim do algoritmo/Programa.
```

Solução 2.3

```
% algoritmo/programa para somar os números pares entre 5 e 500
```

```
clc;
```

```
soma = 0;
```

```
num = 6;
```

```
while ( num <= 500 )
```

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

```
    num = num + 2;
```

```
end
```

```
printf ( 'A soma dos números pares entre 5 e 500 eh %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.**

Solução exercício.

```
% algoritmo/programa para operações elementares sobre dois números
```

```
clc;
```

```
opcao = 0;
```

```
while ( opcao != 5 )
```

```
    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 opção: ");
```

```
    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' );
```

```
    otherwise
```

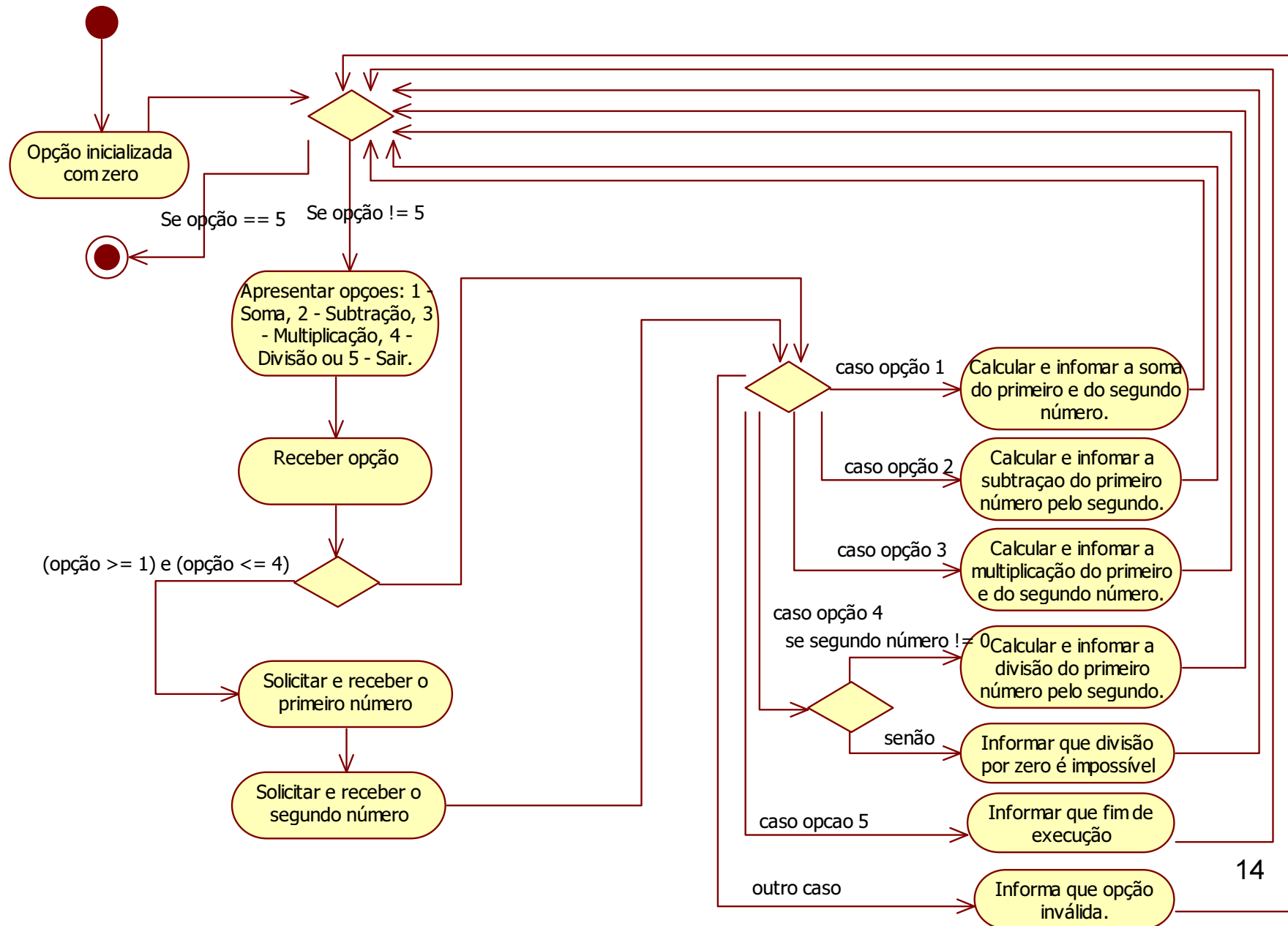
```
        printf ( 'Opção inválida. \n' );
```

```
end
```

```
printf ( '\n \n' );
```

```
end
```

“Fluxograma” (Diagrama de Atividades) do exercício



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.

Exercícios - 5

- **5.1 - Algoritmo para o cálculo do quadrado e da raiz quadrada de 1 até um número dado pelo usuário.**
- **5.2 Algoritmo para o cálculo do fatorial de um número dado pelo usuário.**
 - **Obs.: Usar a estrutura de repetição *enquanto-faça*.**

Algoritmo Fatorial 1

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

if ( numero > 1 )

    resultado = numero;

    while ( numero != 1 )

        numero = numero - 1;
        resultado = resultado * numero;

    end

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

else

    if ( numero >= 1 )
        printf ( 'O fatorial eh 1. ');
    else
        printf ( 'Número inválido. ');
    end

end

% Fim do algoritmo/programa
```

Número = 5

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

Algoritmo Fatorial 2

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

if ( numero > 1 )

    resultado = numero;

    while ( numero != 1 )

        numero = numero - 1;
        resultado = resultado * numero;

    end

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

else

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

end

% Fim do algoritmo/programa
```

Número = 5

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

Exercícios 6.

- **Algoritmo para permitir ao usuário escolher entre o cálculo do fatorial, 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.**

Exercícios 7

- **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.
- **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 *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.

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.