Algoritmos

Estrutura de Repetição

“faça até”

do until

Grupos de Slides No 6.

Prof. SIMÃO

Estrutura de Repetição "repita até"

---

do

conjunto de comandos;

until (*condição ser Verdadeira*)

Obs. : Comando que só funciona em Octave e <u>não</u> em Matlab.

Exemplo de Algoritmo – 1A

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

clc;

num = 1;

do

    printf ( ' Número %d: \n ', num);

    num = num + 1;

until ( num > 1000 )

% fim do algoritmo/programa.
```

Obs. O conjunto de comandos sempre executa pelo menos uma vez... diferentemente do *while*

do until X while

```
clc;

num = 1;

do

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

    num = num + 1;

until ( num > 1000 )
```

```
clc;

num = 1;

while ( num <= 1000 )

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

    num = num + 1;

end
```

As condições são diferentes porque a semântica (ou significado) do *do until* é diferente da semântica do *while*.

Exemplo de Algoritmo - 2

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

clc;

soma = 0;
num = 1;

do

    soma = soma + num;

    num = num + 1;

until ( num > 1000 )

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

% fim do algoritmo/programa.
```

Explicando o Exemplo

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

clc;

soma = 0;
num = 1;

do

    soma = soma + num;

    num = num + 1;

until ( num > 1000 )


printf ( ' O somatório dos números entre 1 e 1000 eh: %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 *do until*.**

Solução Exercício 1

```
% Algoritmo/Programa para imprimir os números de 1 a n.
clc;

soma = 0;

num = 1;

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

fim = input( ' Informe um número: ');

do

    soma = soma + num;

    num = num + 1;

until ( num > fim )

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

% Fim do algoritmo/programa.
```

Exercício 2

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

Obs. Utilizar a estrutura *repita ate*.

Solução Ex.2

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

num = 1;

bandeira = 1;

do

    if ( bandeira )

        printf ( ' O número %d é impar! \n ', num);
        bandeira = 0;

    else

        printf( ' O número %d é par! \n ', num);
        bandeira = 1;

    end

    num = num + 1;

until ( num > 500 )

% fim do algoritmo/Programa.
```

Solução Ex. 2 - V.2

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

num = 1;

bandeira = 1;

do

    if ( bandeira )
       printf ( ' O numero %d é impar! \n ', num);
       bandeira = 0;
    else
       printf ( ' O numero %d é par! \n ' , num);
       bandeira = 1;
    end

    num = num + 1;

until ( num > 500 )

% fim do algoritmo/Programa.
```

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

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 *do until*.

Solução exercício.

```
% algoritmo/programa para operações elementares sobre dois números
clc;
opcao = 0;
do
    printf ( ' Operações elementares sobre dois nuhmeros. \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 ‘ );
until ( opcao == 5 )
```

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 *do until*.

Solução 4.1

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

cont = 1;
nota = 0;
soma = 0;

quantidade = 150;

do

    printf ( ' Informe a %da nota: ' , cont);
    nota = input ( ' ' );

    soma = soma + nota;

    cont = cont + 1;

until ( cont > quantidade )

media = soma /quantidade;

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

% fim do algoritmo/Programa.
```

```
% Algoritmo/Programa para calcular a média de 150 notas.
clc;

cont = 1;
nota = 0;
soma = 0;
quantidade = 150;

do
    do
       flag = 1;
       printf ( ' Informe a %da nota: ' , cont);
       nota = input("");

       if  ( ( nota < 0 ) || ( nota > 10 ) )
          printf ( ' Nota não válida. \n ' );
          flag = 0;
       end

    until (flag == 1)

    soma = soma + nota;

    cont = cont + 1;

until ( cont > quantidade )

media = soma /quantidade;
printf( ' O valor da média eh: %f. \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 end)* para tratar a opção do usuário.**
  - **Para cada uma das versões anteriores usar a estrutura *repita-até (do until)*, comparando com as soluções realizadas anteriormente com a estrutura *equanto (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 tipo *caracter* para tratar a opção do usuário.**
  - **O algoritmo só terminará quando o usuário escolher uma opção de término.**
  - **Utilizar a estrura *repita-até (do until)*.**

- **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 *repita-até (do until)*.**

Exercícios - 7

- **7.1 - Algoritmo para o cálculo do quadrado e da raiz quadrada de 1 até 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 *repita - até*.**

Algoritmo Fatorial – V1

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

if  ( numero > 1 )

   resultado = numero;

   do
      resultado = resultado * ( numero - 1 );

      numero = numero - 1;

   until ( numero == 1 )


   printf ( ' O fatorial é %d: \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
0	5	5
1	5*4=20	4
2	20*3=60	3
3	60*2=120	2
4	120*1=120	1

Algoritmo Fatorial V2

```
% 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 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
```

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		

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.