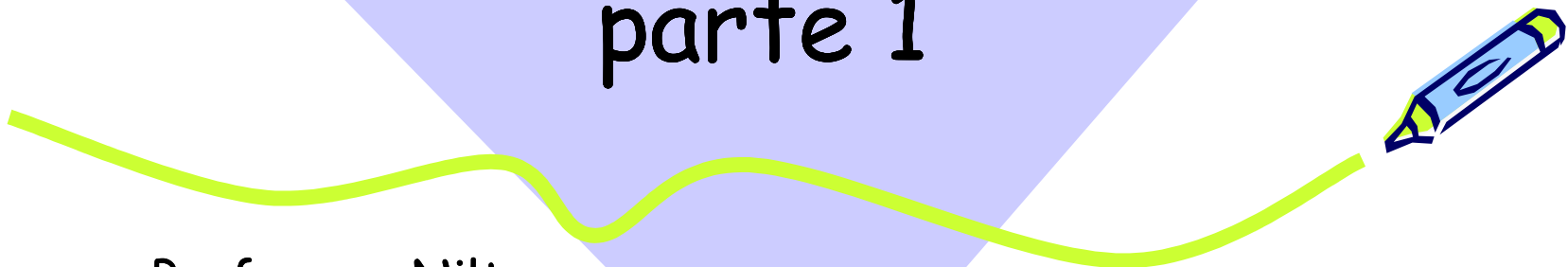


UNIVERSIDADE ESTADUAL DE MATO GROSSO DO SUL  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO



# Disciplina: AEDI Modularização

## parte 1



Professor: Nilton

E-mail: [nilton@comp.uems.br](mailto:nilton@comp.uems.br)

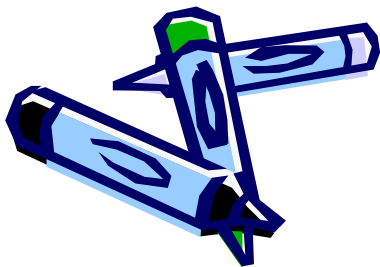
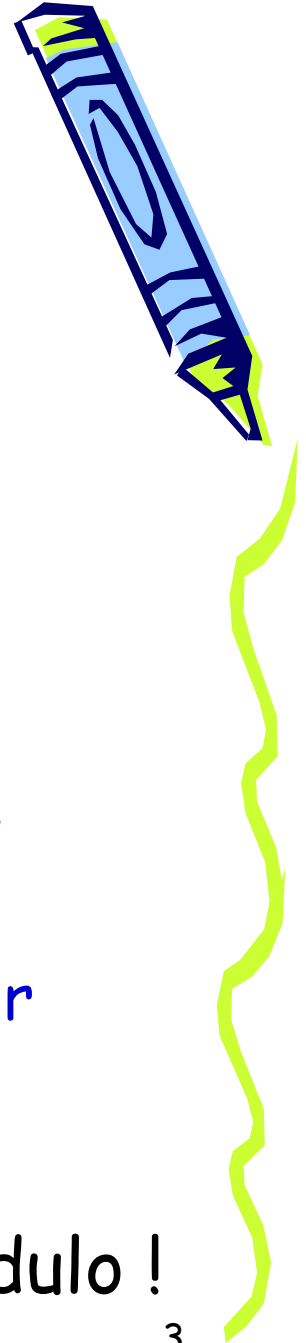
# Modularização

- Problemas complexos exigem algoritmos extensos.
- Dividindo o problema em problemas menores, pode-se achar a solução de cada um dos sub-problemas de forma mais simples.
- A divisão chama-se modularização.



# Modularização

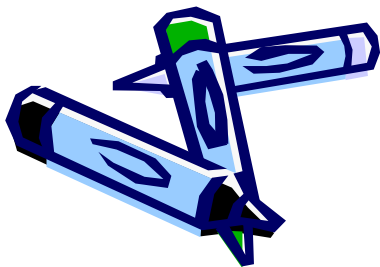
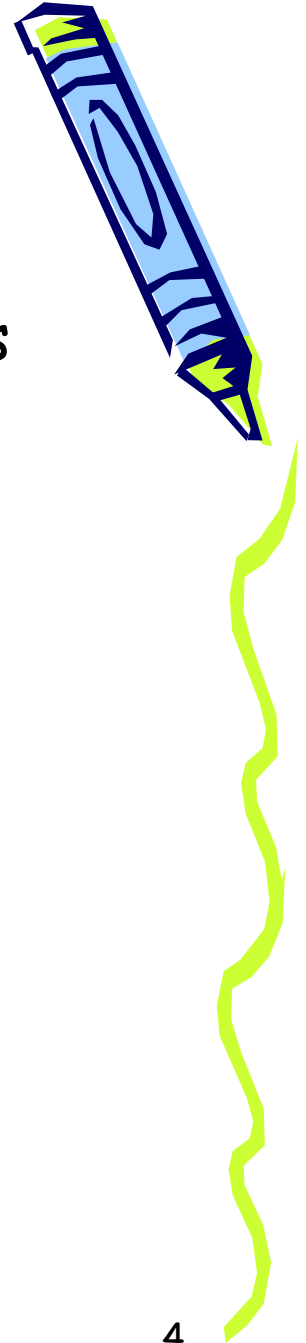
- Algoritmo principal
  - inicia a execução e pode chamar sub-algoritmos.
- Sub-algoritmos
  - um algoritmo que resolve, na maioria das vezes, um problema menor.
  - é chamado pelo algoritmo principal ou por outros sub-algoritmos.



Tudo é módulo !

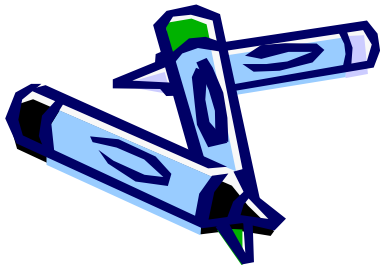
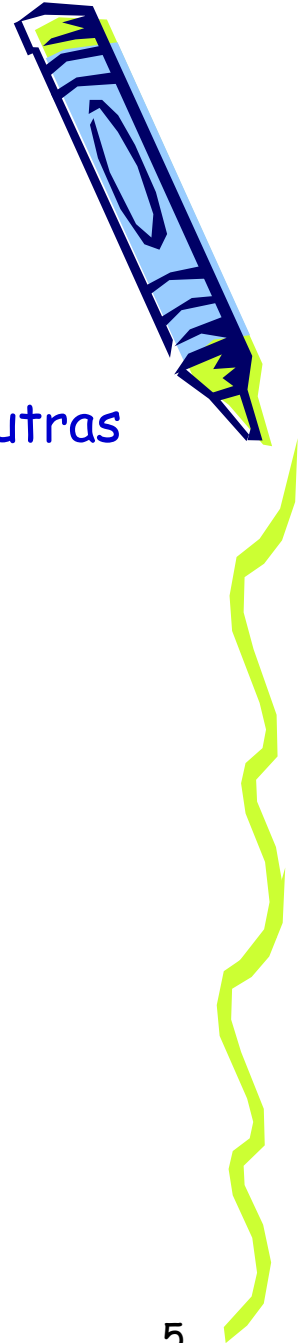
# Modularização: vantagens

- O problema é solucionado através de problemas menores de baixa complexidade
  - Quantidade reduzida de instruções e variáveis
- Reuso de código
- Facilita os testes
- Dividir o desenvolvimento em uma equipe
- Melhora o controle de manutenção.



# Como modularizar

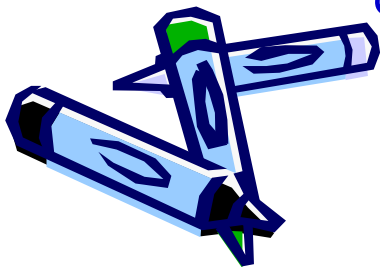
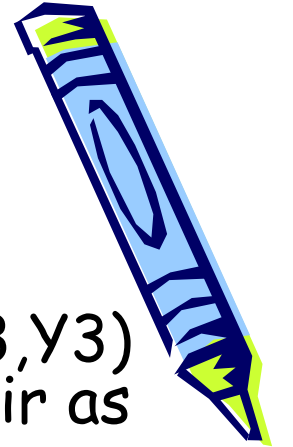
- Identificar a tarefa de cada módulo
  - se a tarefa continua ainda complexa, dividi-la em outras tarefas (*refinamentos sucessivos*)
- Determinar o tipo de módulo
  - procedimento ou função
- Determinar os parâmetros de um módulo
  - Entrada e saída
- Determinar a ligação entre os módulos.



# Um exemplo

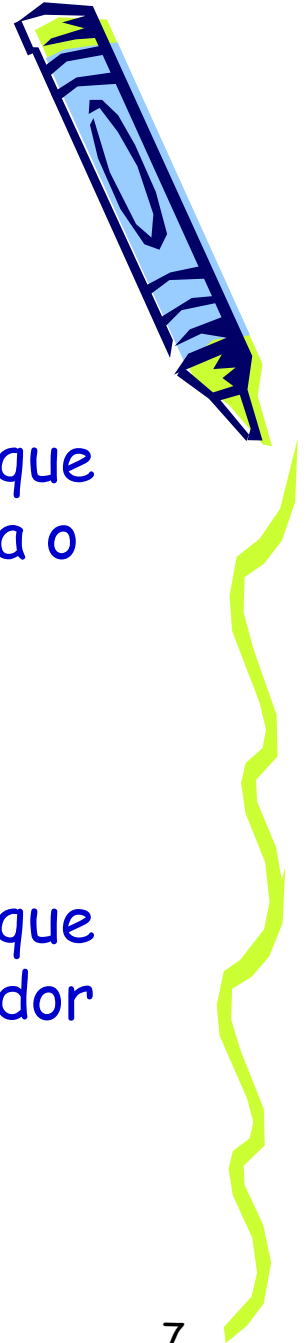
- Dadas as coordenadas (X1,Y1), (X2,Y2) e (X3,Y3) de um triângulo, determine sua área. Imprimir as coordenadas lidas e a área calculada.
- Identificando as principais tarefas
  1. Leitura das coordenadas dos pontos
  2. Cálculo das medidas dos lados do triângulo
  3. Cálculo da área
  4. Escrever as coordenadas lidas e a área calculada
- Refinando a tarefa 2
  - Um lado do triângulo é calculado aplicando-se o cálculo da distância entre dois pontos A, B

$$D_{ab} = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$$



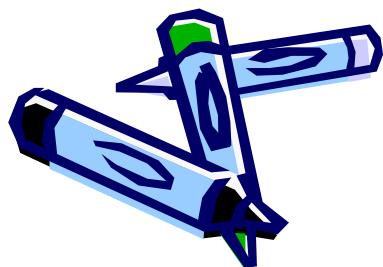
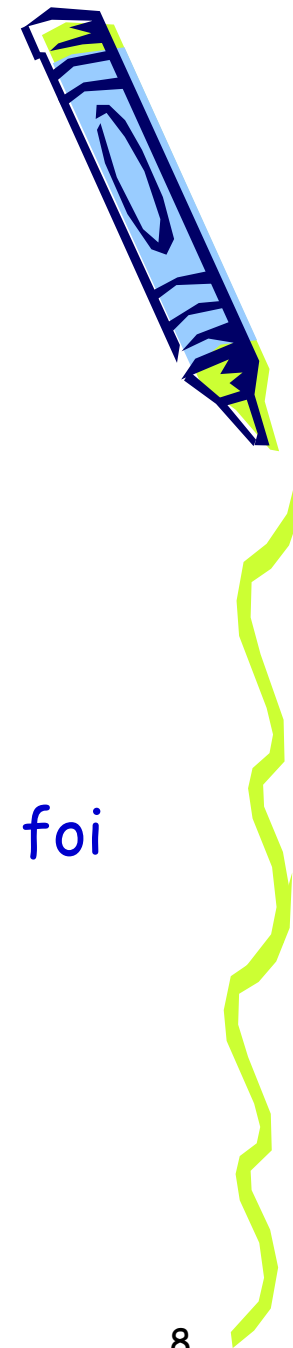
# Tipos de módulos

- Função
  - É um bloco de instruções (sub-algoritmo) que realiza uma tarefa e retorna um valor para o bloco que a chamou.
- Procedimento
  - É um bloco de instruções (sub-algoritmo) que realiza uma tarefa e volta ao bloco chamador sem devolver algum valor específico.



# Tipos de variáveis

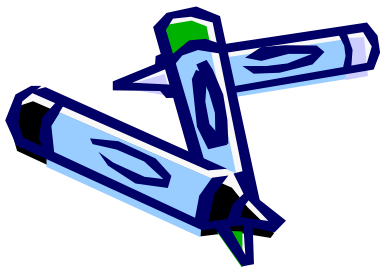
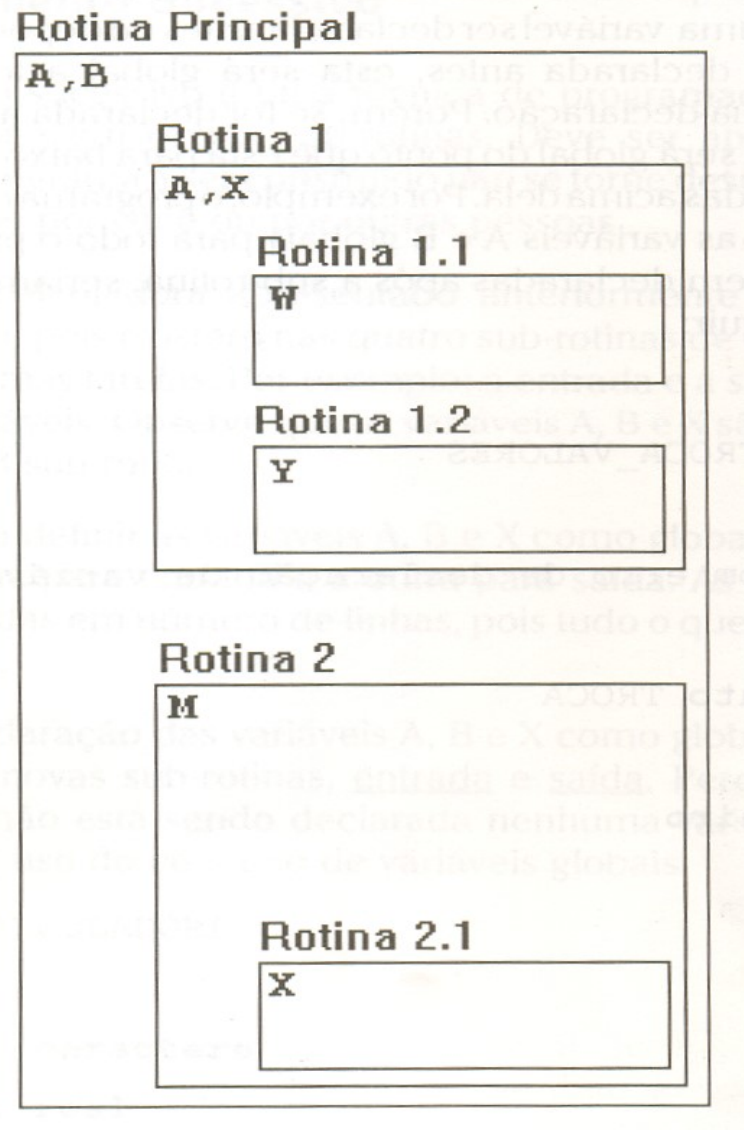
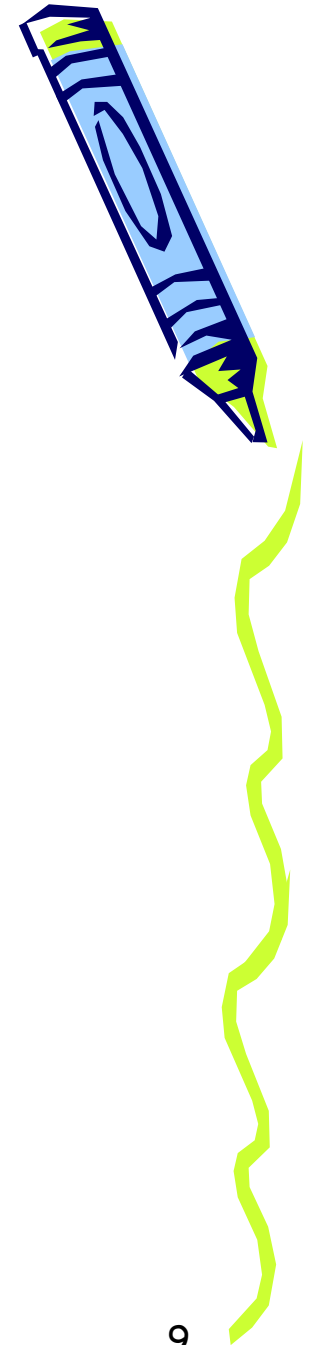
- Variável Global
  - tem visibilidade para o módulo principal e quaisquer outros módulos existentes
- Variável Local
  - Tem visibilidade apenas no módulo no qual foi declarada





# Tipos de variáveis

## escopo



# Criando uma função: sintaxe

**funcao** NOME\_FUNC (<decl\_parametros> ) : **tipo\_basico**

**var**

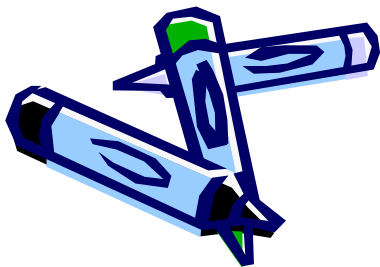
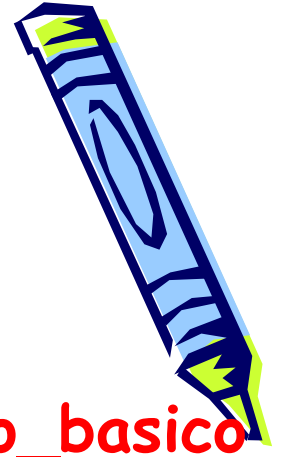
<declaracao variaveis locais>

**inicio**

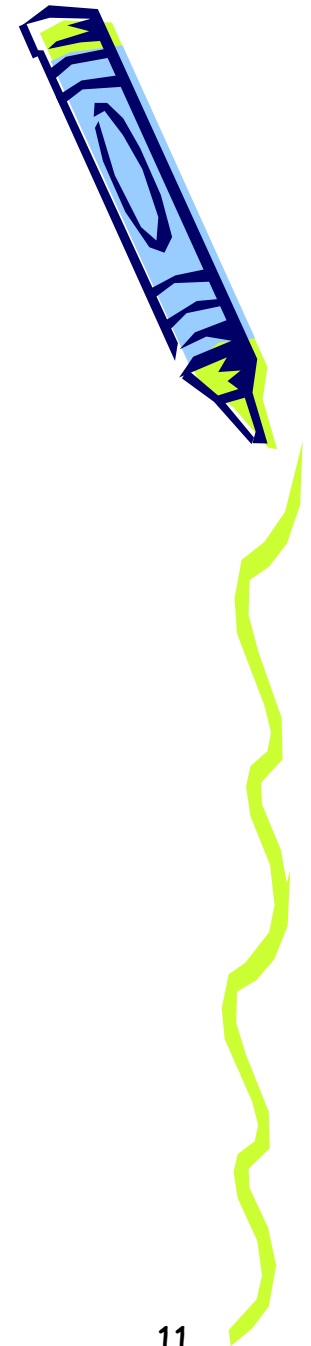
<comandos>

**retorne**(<resultado>)

**fimfuncao**



# Criando uma função: exemplo



```
1 algoritmo "Valor.absoluto.numero"
2 var
3     a, b, x : real
4
5 funcao VALOR_ABS (X:real) :real
6 var
7     ret:real
8 inicio
9     se x>=0 entao
10         ret <- x
11     senao
12         ret <- x * (-1)
13     fimse
14     retorne(ret)
15 fimfuncao
16
17 inicio
18     leia(a)
19     leia(b)
20     escreva(VALOR_ABS(a), VALOR_ABS(b))
21     x <- VALOR_ABS(a * a * a) * a + b
22     escreva(x)
23 fimalgoritmo
```

# Criando uma função: exemplo C/C++

```
1  #include <iostream>
2
3  //declaracao de variaveis globais
4  float a, b, x;
5
6  //funcao ABS
7  float ABS(float x) {
8      //declaracao de variaveis locais
9      float ret;
10
11     //inicio rotina
12     if( x >= 0 )
13         ret=x;
14     else
15         ret=x*(-1);
16     return( ret );
17     //termino rotina
18 }
19
20 //rotina principal
21 int main()
22 {
23     printf("informe o primeiro valor: ");
24     scanf("%f",&a);
25     printf("informe o segundo valor: ");
26     scanf("%f",&b);
27     printf("Os valores absolutos valem:%f          e:%f",ABS(a),ABS(b));
28     x=ABS(a*a*a)*a+b;
29     printf("\nX vale:%f",ABS(x));
30
31     return( 0 );
32 }
```

# Exercícios - função

1- Fazer uma função para calcular o fatorial de um número qualquer.

2- Fazer uma função para calcular a seguinte expressão:

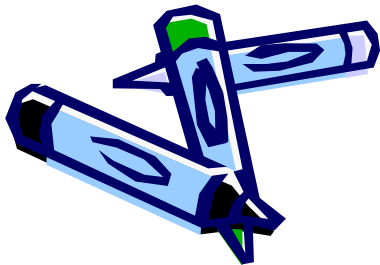
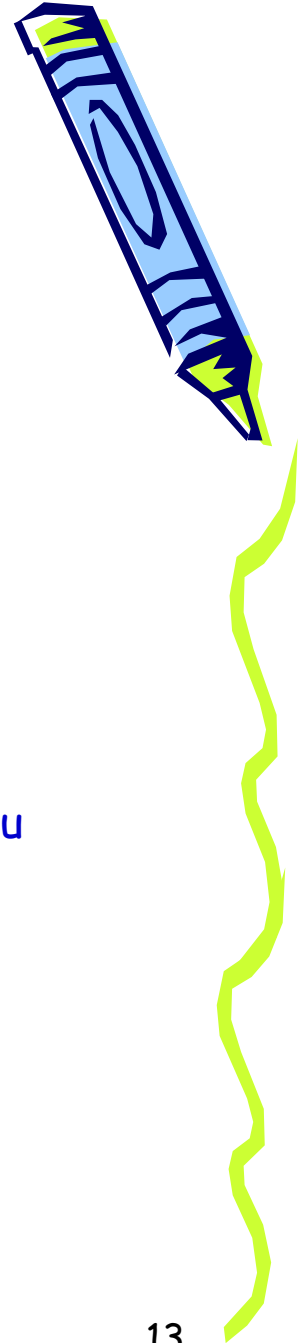
$$S = 1/N + 2/(N-1) + 3/(N-2) + 4/(N-3) + \dots + (N-1)/2 + N/1$$

3- Fazer uma função que verifica se um número é par.

4- Escreva um algoritmo que encontre a soma dos números pares de um conjunto de 1000 números inteiros positivos. Use a função criada do exercício 3.

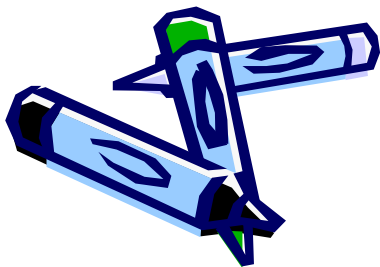
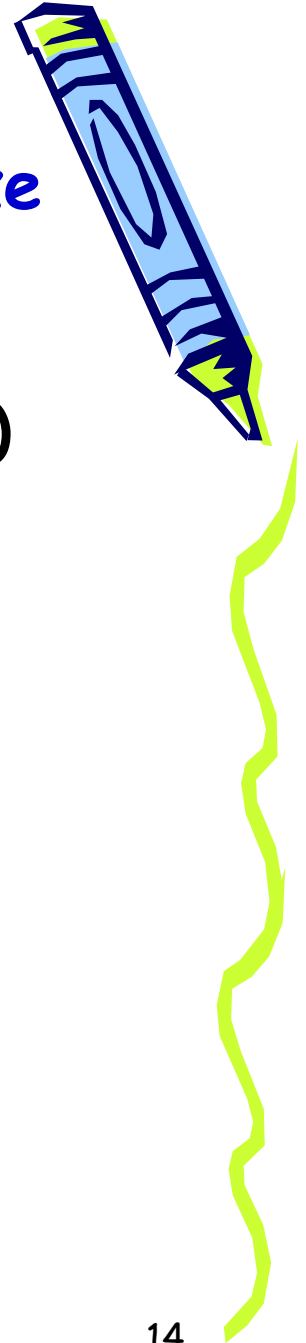
5- Faça um algoritmo que determine o dobro de um número se for par e seu triplo se for ímpar. Crie o máximo de módulos que você achar conveniente.

6- Fazer uma função que verifique se uma senha digitada pelo usuário é válida. A senha válida é "ASDFG".



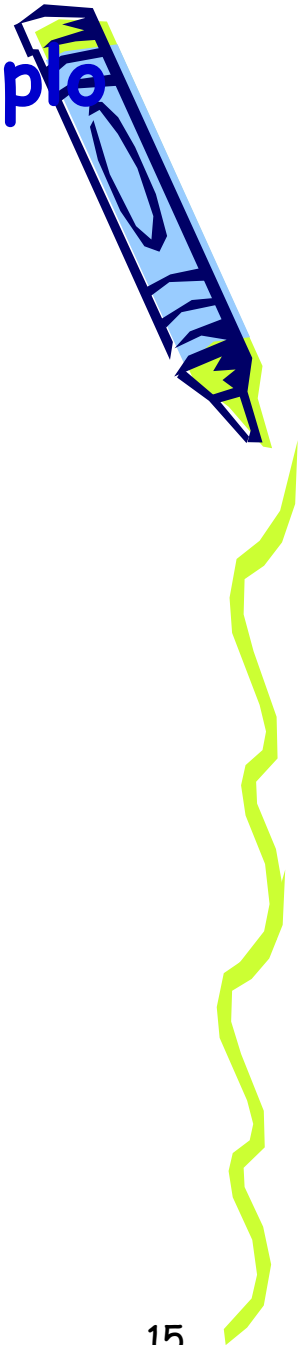
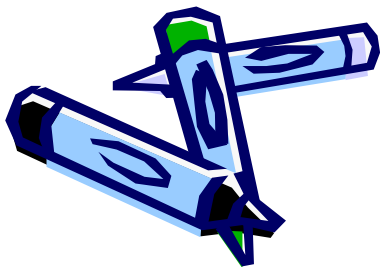
# Criando um procedimento: **sintaxe**

```
procedimento NOME_PROC (<decl_parametros> )  
var  
    <declaracao variaveis locais>  
inicio  
    <comandos>  
fimprocedimento
```

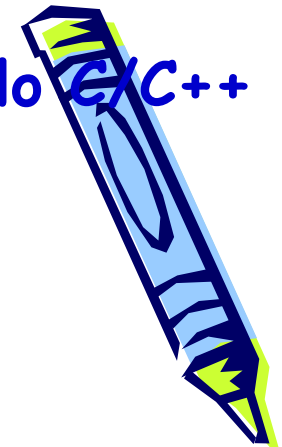


# Criando um procedimento: exemplo

```
1 algoritmo "troca.valores"  
2 var  
3   a, b: inteiro  
4  
5 procedimento TROCA  
6 var  
7   aux: inteiro  
8 inicio  
9   aux ← a  
10  a ← b  
11  b ← aux  
12 fimprocedimento  
13  
14 inicio  
15   leia(a)  
16   leia(b)  
17   TROCA  
18   escreval(a,b)  
19   leia(a)  
20   leia(b)  
21   TROCA  
22   escreval(a,b)  
23 fimalgoritmo
```



# Criando um procedimento: exemplo C/C++

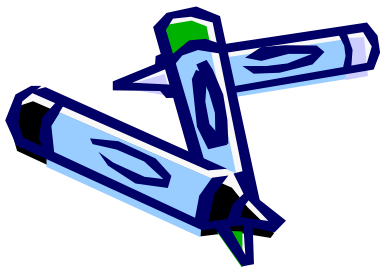
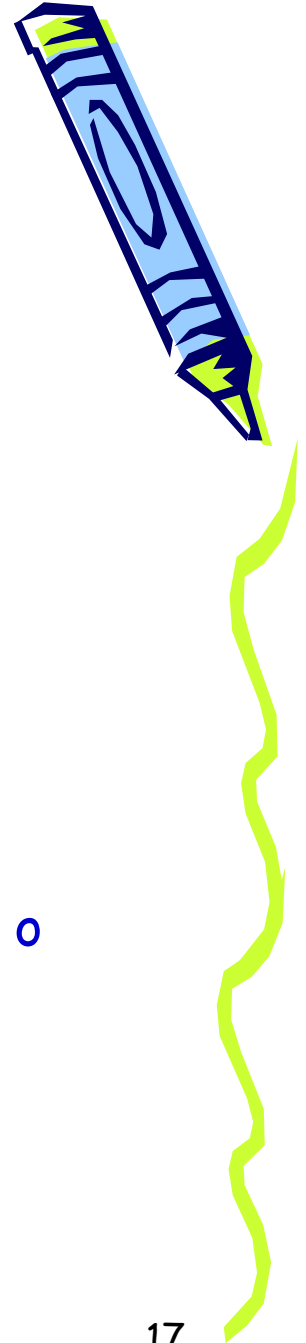


```
1  #include <iostream>
2
3  //declaracao de variaveis globais
4  int a, b;
5
6  //funcao ABS
7  void TROCA() {
8      //declaracao de variaveis locais
9      int aux;
10
11     //inicio rotina
12     aux = a;
13     a = b;
14     b = aux;
15     //termino rotina
16 }
17
18 //rotina principal
19 int main()
20 {
21     printf("\nInforme A: ");
22     scanf("%d",&a);
23     printf("\nInforme B: ");
24     scanf("%d",&b);
25     TROCA();
26     printf("\n%d %d",a,b);
27     printf("\nInforme A: ");
28     scanf("%d",&a);
29     printf("\nInforme B: ");
30     scanf("%d",&b);
31     TROCA();
32     printf("\n%d %d",a,b);
33
34     return( 0 );
35 }
```



# Exercícios - procedimento

- 1- Faça um procedimento que coloque três valores em ordem crescente.
- 2- Faça um procedimento para mostrar um menu de opções ao usuário, como abaixo:
  - (a) inserir
  - (b) remover
  - (c) pesquisar por nome
  - (d) pesquisar por idade
  - (e) sair
- 3- Faça uma função para ler a opção desejada pelo usuário para o exercício 2.

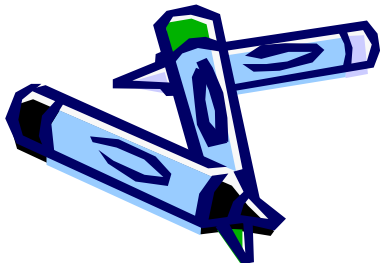
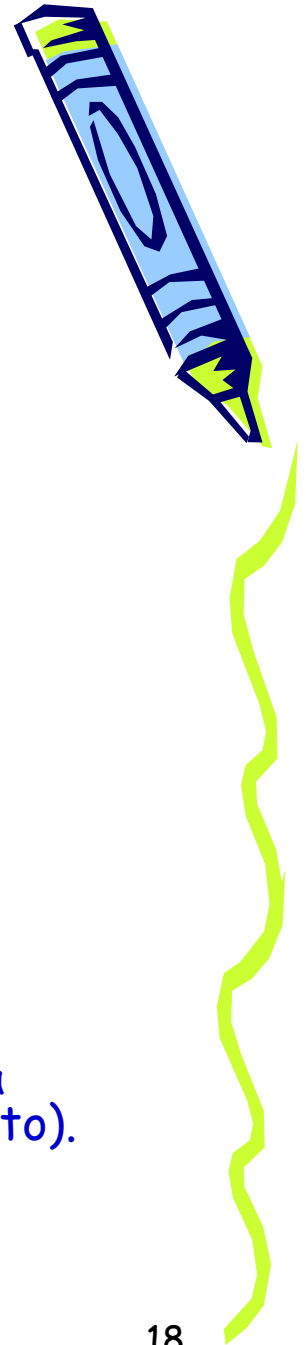


# Exercícios gerais

1-Faça um algoritmo que dada uma matriz  $M \times M$  de números positivos mostre:

- A - O traço da matriz;
- B - Os elementos da diagonal secundária;
- C - A média do traço encontrado no item A;
- D - Os números pertencentes ao traço maiores que a média;
- E - O maior número em cada linha, considerando a inexistência de empates;
- F - O maior número em cada coluna, considerando a inexistência de empates;
- G - A média de cada coluna e linha;
- H - Os números pares;
- I - Os números ímpares;
- J - A matriz simétrica.

O algoritmo deverá dispor de um menu de opções para o usuário e para cada item (A, B, ...) crie uma ou mais rotinas (função/procedimento).

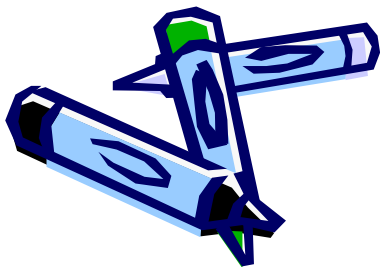


# Exercícios gerais

2- Construir um algoritmo que contenha três rotinas (cálculo do quadrado, cálculo do cubo e múltiplo). Na rotina principal leia um intervalo  $[A, B]$  e imprima entre o intervalo A e B:

- O quadrado de X, se X for múltiplo de 3;
- O cubo de X, se X for múltiplo de 5;
- O próprio X, se X não for múltiplo de 3 e 5.

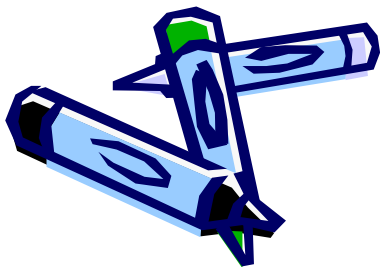
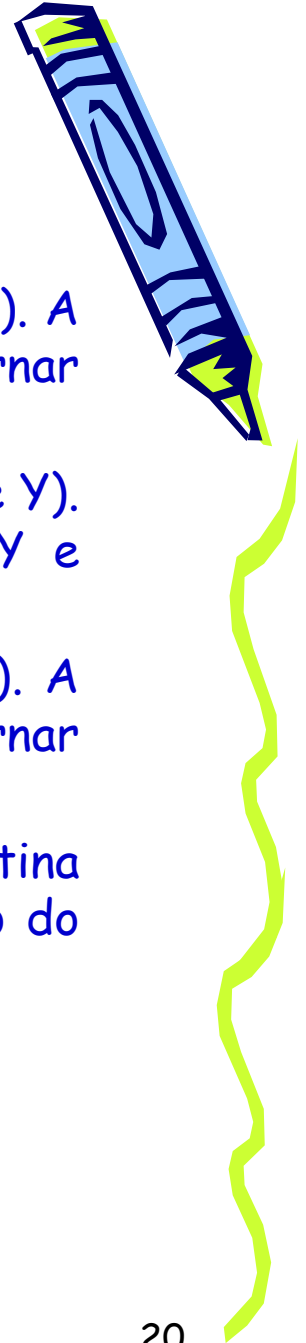
3- Escrever uma rotina que tenha como parâmetros horas, minutos e segundos e devolva o número total de segundos. Escreva o algoritmo que teste a rotina, ou seja, deverá ler os valores, chamar a rotina e mostrar o resultado.



# Exercícios gerais

## 4- Faça rotinas:

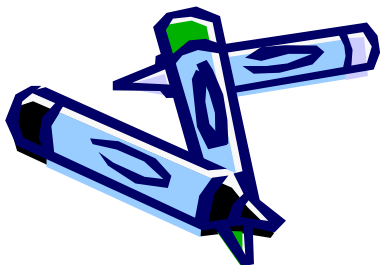
- Uma rotina para calcular a subtração de dois números reais ( $X$  e  $Y$ ). A rotina deve receber, como parâmetros, os valores de  $X$  e  $Y$  e retornar o resultado ( $X-Y$ ).
- Uma rotina para calcular a multiplicação de dois números reais ( $X$  e  $Y$ ). A rotina deve receber, como parâmetros, os valores de  $X$  e  $Y$  e retornar o resultado ( $X*Y$ ).
- Uma rotina para calcular a divisão de dois números reais ( $X$  e  $Y$ ). A rotina deve receber, como parâmetros, os valores de  $X$  e  $Y$  e retornar o resultado ( $X/Y$ ).
- Uma rotina para calcular o fatorial de um número inteiro  $X$ . A rotina deve receber, como parâmetro, o valor de  $X$  e retornar o cálculo do fatorial.



# Exercícios gerais

- 5- Escreva um algoritmo que deve ler dois números inteiros positivos (M e P), calcular (usando algumas rotinas do exercício anterior e construindo outras) e escrever o número de arranjos e combinações de M elementos P a P, dado pelas fórmulas:

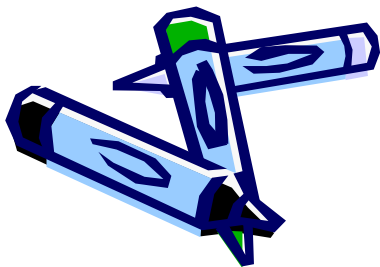
$$A_M^P = \frac{M!}{(M-P)!} \quad C_M^P = \frac{M!}{P!(M-P)!}$$



# Exercícios gerais

6- Faça as rotinas:

- Uma rotina para ler um ano correto, usando o protótipo:  
função ler\_ano():inteiro;  
*(considere que o valor válido para ano está entre 1984 e 2009)*
- Uma rotina para ler um mês correto, usando o protótipo:  
função ler\_mes():inteiro;
- Uma rotina para ler um dia correto, usando o protótipo:  
função ler\_dia(mês, ano:inteiro):inteiro;



# Exercícios gerais

7- A avaliação de aproveitamento de uma certa disciplina é feita através de 2 provas mensais no valor de 30 pontos cada e uma prova final no valor de 40 pontos. A nota final é obtida somando-se as notas das provas. O conceito final é dado atendendo-se ao seguinte critério:

De 90 a 100 pontos - conceito A

De 80 a 89 pontos - conceito B

De 70 a 79 pontos - conceito C

De 60 a 69 pontos - conceito D

De 40 a 59 pontos - conceito E

De 0 a 39 pontos - conceito F

Fazer uma rotina que, recebendo como parâmetros as notas das três provas, devolva o conceito calculado como explicado acima.

