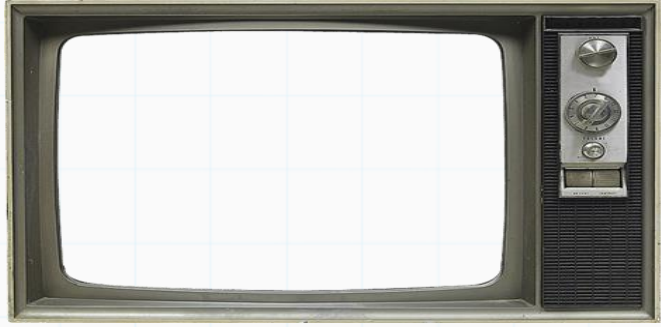
Programação De Computadores



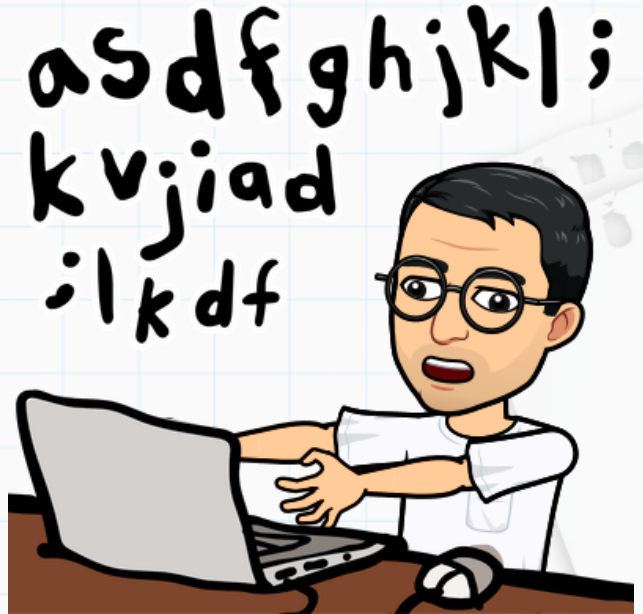
Professor : Yuri Frota

www.ic.uff.br/~yuri/prog.html

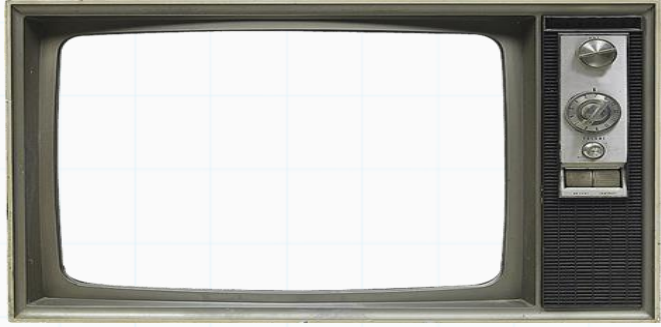
yuri@ic.uff.br

```
int soma(int A, int B)
{
    int C;
    C = A + B;
    return(C);
}
```

```
float calculaMedia(int *v) {

    int i, soma=0;
    for (i=0; i<TAM; i++)
        soma += v[i];

    return (soma/(TAM*1.0));
}
```

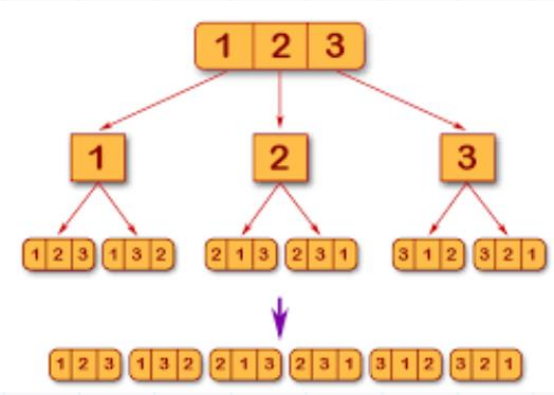


Funções- LAB



1) Permutação: Um número N1 é dito uma permutação de um outro número N2 se os dígitos de N1 formam uma permutação dos dígitos de N2.

Exemplo: 5412434 é uma permutação de 4321445, mas não é uma permutação de 4312455.



Obs.: Considere que o dígito 0 (zero) não aparece nos números.
Obs2: O programa deve ser feito apenas com variáveis numéricas unidimensionais (ex: int, float, etc... não pode vetores e matrizes)

(a) Faça uma função contadígitos que dados um inteiro N e um inteiro d, 0 < d <= 9, devolve quantas vezes o dígito d aparece em n.

(b) Usando a função do item anterior contadígitos, faça um programa que lê dois inteiros positivos N1 e N2 e responda se N1 é permutação de N2. Para isso basta checar se para cada dígito (de 1 até 9) ele aparece o mesmo número de vezes em N1 e N2, se não aparecer, não é permutação.

Exemplo:

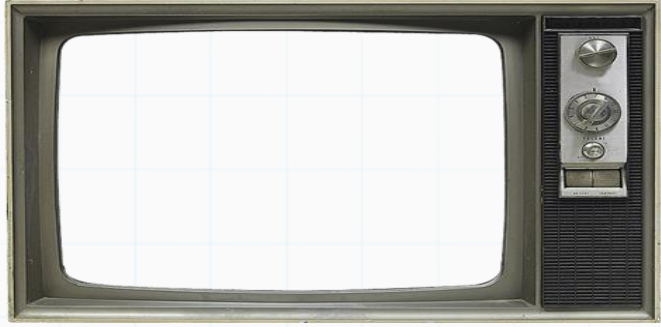
```
Digite dois inteiros: 5412434 4321445
5412434 e' permutacao de 4321445

Digite dois inteiros: 1113 3111
1113 e' permutacao de 3111

Digite dois inteiros: 5555 555
5555 nao e' permutacao de 555
```

Lembre-se:

int x;

x%10; // ultimo dígito de x
x/10; // x sem o último digito

Funções- LAB



2) Ordenação: Dado um vetor de inteiros V de tamanho n>0, e um inteiro k<n, escreva uma função:

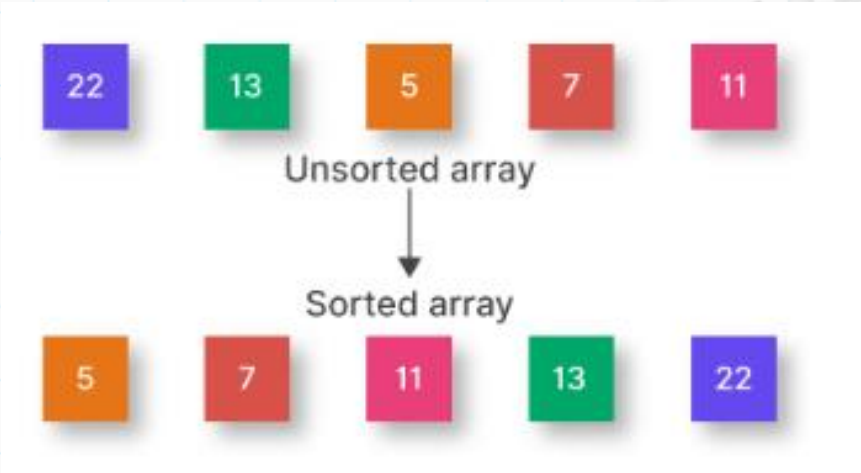
```
int indice_min(int n, int v[n], int k)
```

que retorne o índice do menor elemento entre V[k], V[k+1], ..., V[n-1],
usando essa função, faça uma outra função que ordene o vetor de forma crescente

```
void ordena(int n, int v[n])
```

Usando essas duas funções faça um programa para ordenar o vetor da seguinte forma:

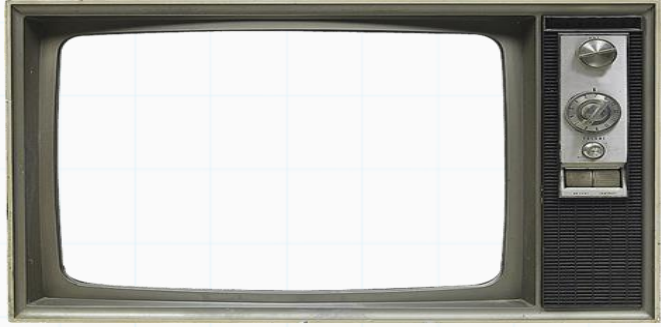
Percorra o vetor do começo até o fim (de 0 até n-1), e para cada posição k, encontre o índice do menor elemento de k até o fim, e troque os elementos entre k e o de menor elemento. No fim do laço o vetor deverá estar ordenado.



Exemplo:

```
n: 8
vetor = 4, 9, 7, 2, 1, 8, 15, 88,
vetor ordenado = 1, 2, 4, 7, 8, 9, 15, 88,
```

Funções- LAB



3) Quadrado Latino: Dizemos que uma matriz A nxn é um quadrado latino de ordem n se em cada linha e em cada coluna aparecem todos os inteiros 1,2,3,...,n (ou seja, cada linha e coluna é permutação dos inteiros 1,2,...,n).

Exemplo:

1	2	3	4
2	3	4	1
4	1	2	3
3	4	1	2

- Faça funções de ler e imprimir a matriz

void lematriz (int n, int m, int mat[n][m])
void imprimematriz (int n, int m, int mat[n][m])

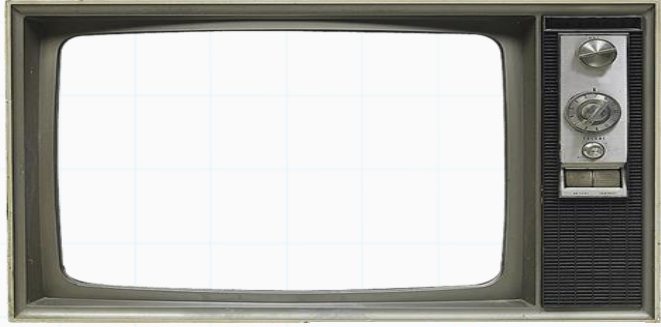
- Faça uma função uma função que dado uma matriz, diga se essa matriz é latina ou não

int latina (int n, int m, int mat[n][m])

- Use essas funções para fazer o programa

Dica: Vamos ter que usar laços triplos

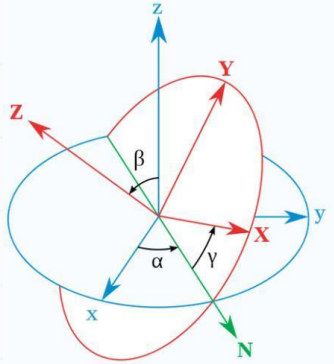
Funções- LAB



4) Ângulos: Faça uma função arctan que recebe o número real 0<=x<=1 e devolve uma aproximação do arco tangente de x (em radianos) através da série:

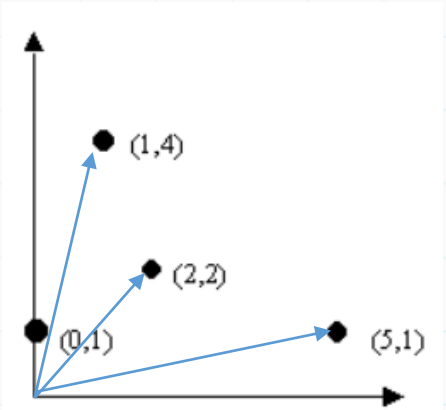
$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \ldots$$

incluindo todos os termos da série enquanto $\left|\frac{x^k}{k}\right| > 0.0001$



Faça depois uma outra função chamada angulo que receba um ponto cartesiano real (x,y) do primeiro quadrante estrito (x>0 ou y >0) e devolve o ângulo (em graus) formado entre o vetor e o eixo horizontal . Exemplo:

(0,1)	90 graus
(2,2)	45 graus
(1,4)	75 graus
(5,1)	11 graus



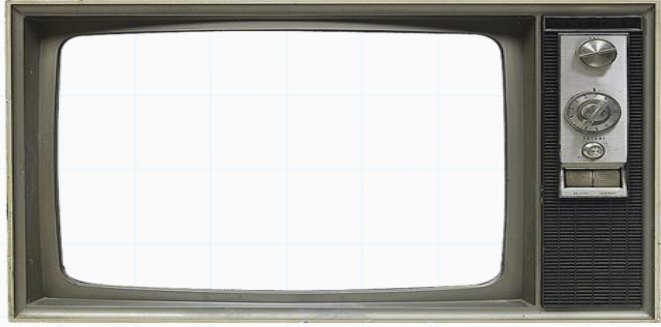
Para calcular esse ângulo α, usamos a formula:

$$\alpha = \begin{cases} \arctan\left(\frac{y}{x}\right), caso\ y < x \\ \frac{\pi}{2} - \arctan\left(\frac{x}{y}\right), caso\ contrário \end{cases}$$

A formula dará o ângulo em radianos, não esqueça de converter para graus (multiplique por 180/ π) usando um valor de π de 3.14.

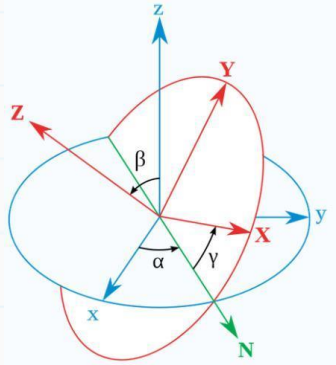
Com essas duas funções faça um programa que leia n pontos no primeiro quadrante estrito e diga qual o que forma menor ângulo com o eixo horizontal. Vamos ver a seguir um exemplo de execução:

Funções- LAB

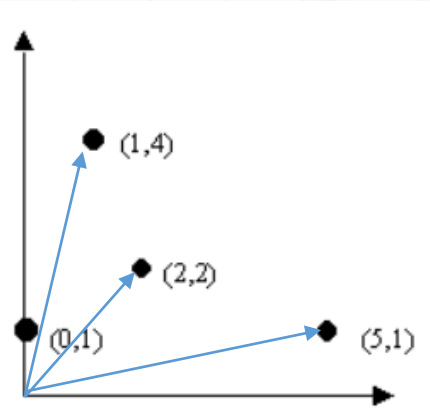


Exemplo:

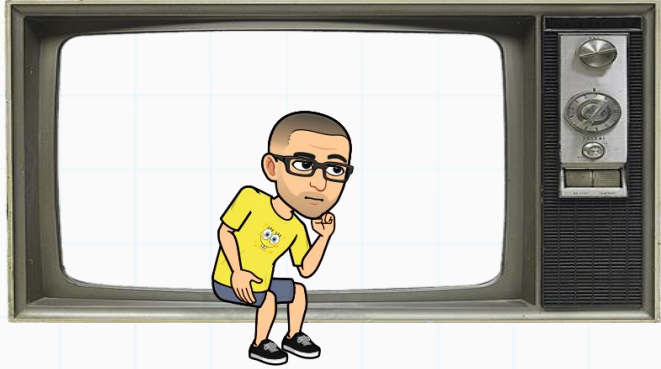
```
Digite quantidade de pontos: 4

Digite x e y: 0 1
PI/2 - arctan(x/y)=0.000000
angulo = 90.000000

Digite x e y: 2 2
arctan(y/x)=0.785349
angulo = 45.020023

Digite x e y: 1 4
PI/2 - arctan(x/y)=0.244987
angulo = 75.956161

Digite x e y: 5 1
arctan(y/x)=0.197333
angulo = 11.312102

O ponto de menor angulo: (5.000000,1.000000)
Menor angulo: 11.312102 graus
```



(0,1)	90 graus
(2,2)	45 graus
(1,4)	75 graus
(5,1)	11 graus



Funções- LAB



5) Matriz neperiana: dada uma matriz de reais quadrada A com dimensão nxn, e um inteiro k>0, definimos uma matriz neperiana e^A como:

$$e^A = I_n + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \frac{A^4}{4!} + \cdots \frac{A^k}{k!}$$

onde In define a matriz identidade de cardinalidade n. Para fazer essa questão, implemente primeiro:

a) Uma função que recebe como parâmetros um inteiro n e duas matrizes quadradas reais A e B de ordem n. Esta função devolve em uma matriz AB=A+B, também passada como parâmetro.

void soma(int n, float A[n][n], float B[n][n], float AB[n][n])

b) Uma função que recebe como parâmetro um número inteiro n, um número real val e uma matriz Anxn. A função multiplica a matriz A pelo número val.

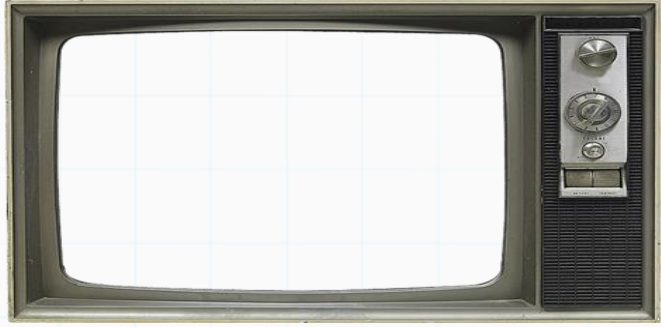
void multiplica_real(int n, float A[n][n], float val)

c) Uma função que recebe como parâmetros um inteiro n e duas matrizes quadradas reais Anxn e Bnxn. Esta função devolve em uma matriz AB, também passada como parâmetro, o produto das matrizes A e B.

void multiplica(int n, float A[n][n], float B[n][n], float AB[n][n])

Pense em quais outras funções você vai precisar para implementar também o programa. O programa deve receber inteiros n e k, e matriz de reais A nxn e determinar a aproximação de e^A. Veja o exemplo a seguir:

Funções- LAB



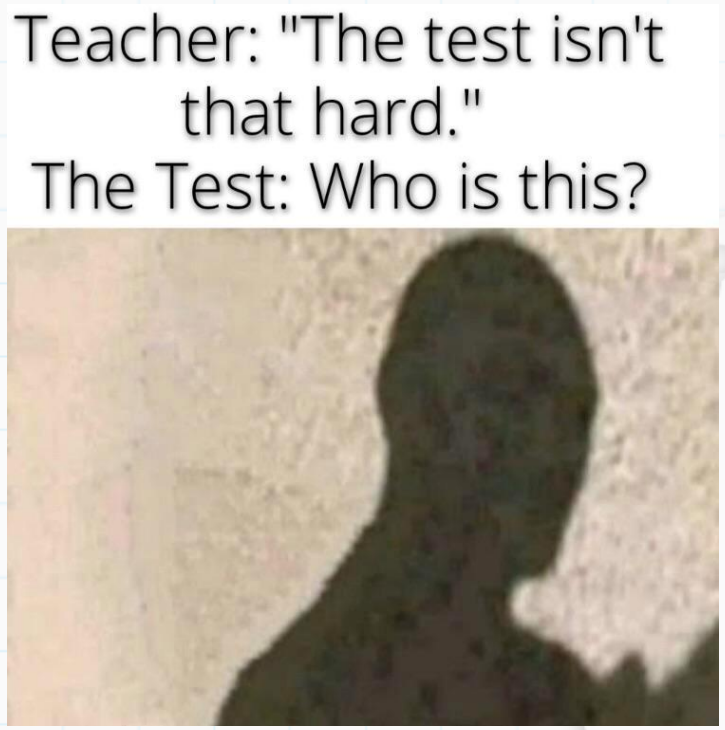
Exemplo:

```
n: 2
k: 4
A[0][0]: 1
A[0][1]: 2
A[1][0]: 3
A[1][1]: 4

matriz A
 1.00  2.00
 3.00  4.00

SOMA = A+I
 2.00  2.00
 3.00  5.00

A2
 7.00  10.00
 15.00  22.00

TERMO = A2 / 2!
 3.50  5.00
 7.50  11.00
```

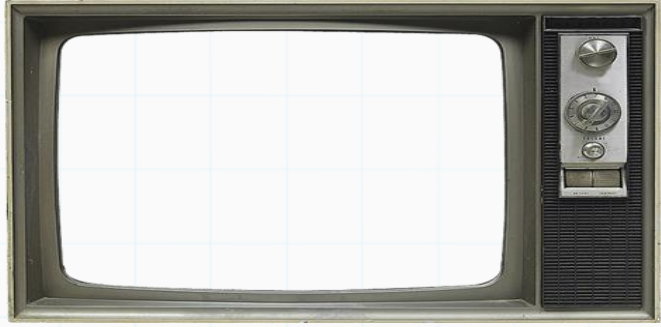
```
SOMA = SOMA + TERMO
 5.50  7.00
 10.50  16.00

A3
 37.00  54.00
 81.00  118.00

TERMO = A3 / 3!
 6.17  9.00
 13.50  19.67

SOMA = SOMA + TERMO
 11.67  16.00
 24.00  35.67

A4
 199.00  290.00
 435.00  634.00

TERMO = A4 / 4!
 8.29  12.08
 18.13  26.42
```

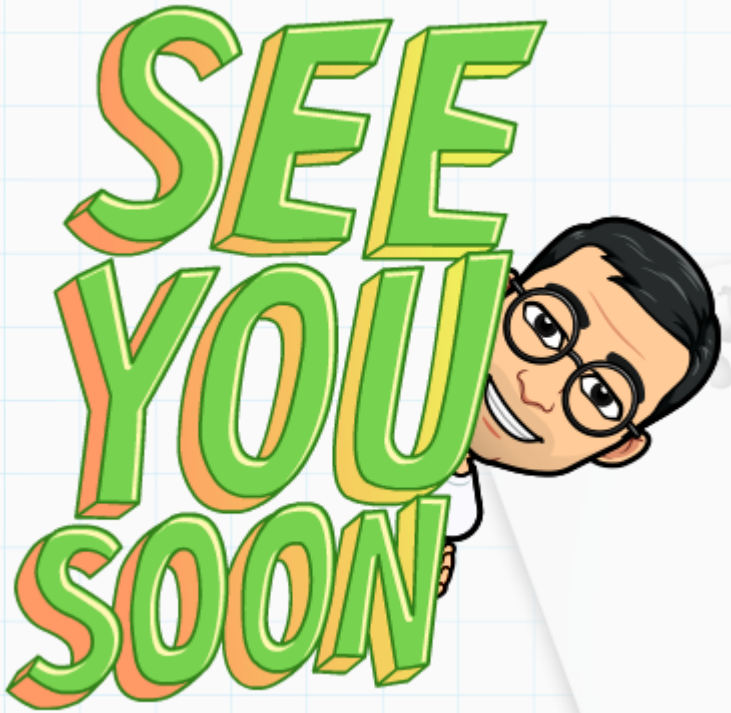
```
SOMA = SOMA + TERMO
 19.96  28.08
 42.13  62.08
```

$$e^A = I_n + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \frac{A^4}{4!} + \cdots \frac{A^k}{k!}$$



Até a próxima





Slides baseados no curso de Aline Nascimento