

Para cada problema, desenvolva o algoritmo, compile e teste. Para os dois primeiros, submeta ao juiz online. A submissão deverá ter recebido **Accepted** e **utilizar as funções implementadas**. Chame o professor para avaliar cada programa assim que estiver pronto e testado. Para conseguir a nota máxima é necessário resolver todos os exercícios em aula.

1. Menor e Posição (USANDO NOTAÇÃO DE PONTEIROS)

Escreva uma função que recebe um inteiro N e um ponteiro para um arranjo com N elementos. A função deve imprimir o menor valor do arranjo e sua posição. Utilize esta função para resolver o seguinte problema: <https://www.urionlinejudge.com.br/judge/pt/problems/view/1180>

2. Linha na Matriz

Implemente as seguintes funções:

- `somaLinha`: recebe uma matrix de inteiros 12×12 e um inteiro L , após retorna a soma dos elementos na linha L .
- `mediaLinha`: recebe uma matrix de inteiros 12×12 e um inteiro L , após retorna a média dos elementos na linha L .

Utilize estas funções para resolver o seguinte problema:

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1181>

3. O ângulo entre dois vetores v e w no plano pode ser calculado usando o produto escalar (produto interno padrão) e a inversa do cosseno conforme a equação:

$$\theta = \arccos\left(\frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \|\mathbf{w}\|}\right)$$

Implemente uma função `angulo` que recebe as coordenadas v_x , v_y , w_x e w_y dos vetores, e retorne o ângulo *theta* entre eles. Veja que uma função também é necessária para calcular o produto escalar. Implemente a função `escalar`, que receberá as coordenadas dos dois vetores e retornará o seu produto escalar. Além disso, implemente a função `radPgauas` para converter um ângulo de radianos para graus.

Implemente um programa que leia as coordenadas dos vetores, e chame a função `angulo`. Note que este ângulo será em radianos, use a função `radPgauas` para fazer a conversão. Ao final, exiba o ângulo em graus e use a função `char s_n()` do exercício anterior para perguntar se o usuário deseja calcular um novo ângulo.

Use `#define PI 3.14159265` e as funções `acos` e `fabs` da `math.h`

Exemplo de execução:

```
Digite as coordenadas de v: 10 0
Digite as coordenadas de w: 0 3
```

```
O angulo entre v e w eh: 90 graus
Mais um (S/N)? S
```

```
Digite as coordenadas de v: 1 0
Digite as coordenadas de w: 300 0
```

```
O angulo entre v e w eh: 0 graus
Mais um (S/N)? G << BEEP >>
Mais um (S/N)? N
```