

# MC202 BC - Estrutura de

## Lab 02 Dados nas Janelas da Matriz

**Prazo de Entrega:** 12/03/2014 - 23:59:59

**Link Susy:** <https://susy.ic.unicamp.br:9999/mc202bc/02>  
(<https://susy.ic.unicamp.br:9999/mc202bc/02>)

As matrizes estão presentes em muitos problemas computacionais e matemáticos. Saber como implementá-las e manipulá-las, o tornará apto a resolver tais problemas.

Considere uma matriz com **r** linhas e **c** colunas. Cada um de seus elementos possui uma posição (x,y), em que  $0 \leq x \leq r-1$  e  $0 \leq y \leq c-1$ .

Sua tarefa será realizar sequências de operações em sub-matrizes ("janelas") desta matriz principal.

Você deverá implementar uma TAD para a resolução deste laboratório. Para facilitar o seu trabalho, estamos disponibilizando os cabeçalhos das funções no arquivo **lab02.h** no Susy (*Arquivos auxiliares*). Você deverá implementar todas as funções desta TAD em um arquivo chamado **lab02.c**, que também conterá a **main** de seu programa.

A não implementação e o não uso de qualquer uma das funções da TAD implicará em **nota ZERO**, independente das saídas do programa.

### Observações Gerais

Salve seu código com o nome: **lab02.c** e insira seu nome e RA nos locais indicados (nunca use acentuação num programa em C):

```
/* Nome: Seu  
nome aqui  
* RA: Seu RA  
aqui  
* Laboratorio 02  
- As Janelas da
```

Matriz \*/

...

SEU

PROGRAMA

## MC202 BC

Laboratórios  
(/~ra144681/courses/mc202/)

(/~ra144681/courses/mc202/labs)

Slides

(/~ra144681/courses/mc202/slides)

Notas

(/~ra144681/courses/mc202/notas)

1. Utilize **return 0;** na main de seu programa;
2. Você deverá implementar todas as funções desta TAD em um arquivo chamado **lab02.c**, que também conterá a **main** de seu programa. A não implementação e o não uso de qualquer uma das funções da TAD implicará em **nota ZERO**, independente das saídas do programa.
3. A matriz principal utilizada deve ser alocada DINAMICAMENTE. A não alocação dinâmica ocasionará em **nota ZERO**, independente das saídas do programa;
4. Todas as bibliotecas necessárias para a utilização do código já estão incluídas no arquivo **lab02.h**. NÃO INCLUA mais nenhuma biblioteca;
5. Na compilação do código no Linux, você pode utilizar: `gcc -o lab02 lab02.c -ldiretorio_do_header`
6. Qualquer tentativa de fraude, plágio e afins, corresponderá em **nota ZERO** para os envolvidos;
7. Códigos ilegíveis serão considerados errados. A legibilidade é obtida com indentação correta e coerente, bons nomes de variáveis e funções, bem como boa subdivisão do código em funções auxiliares;

## Crerios de Avaliao

$$nota = \frac{\sum_i^n c_i}{n} * 10,$$

onde **Ci** é o caso de teste i, com valores 1 (certo) e 0 (errado) e **n** é o número total de casos de teste.

A NÃO implementação e o NÃO uso das funções da TAD **lab02.h** implicará em **nota ZERO**, independente das saídas do programa.

## Entrada

A primeira linha contém 3 valores inteiros positivos **r**, **c**, **m**, em que  $m$  ( $m \geq 1$ ) é o número de operações a serem realizadas.

Cada uma das próximas **m** linhas de entrada, conterá uma das operações mostradas a seguir:

1  $x_1$   $y_1$   $x_2$   $y_2$   $v$

Incrementa o valor inteiro **v** em cada elemento **(x,y)** da sub-matriz **(x1,y1,x2,y2)**. Utilize a função *increment\_matrix*.

2  $x_1$   $y_1$   $x_2$   $y_2$   $v$

Atribui o valor inteiro **v** em cada elemento **(x,y)** da sub-matriz **(x1,y1,x2,y2)**. Utilize a função *set\_value\_into\_matrix*.

3  $x_1$   $y_1$   $x_2$   $y_2$

Imprime o valor do **somatório**, **mínimo** e **máximo** dos elementos da sub-matriz **(x1,y1,x2,y2)**. Utilize a função *print\_matrix\_metrics*.

\* Nas operações descritas acima, a sub-matriz **(x1,y1,x2,y2)** significa que todos os elementos **(x,y)** satisfazem  $x_1 \leq x \leq x_2$  e  $y_1 \leq y \leq y_2$ .

## Saída

A saída é acionada pelo seguinte comando de entrada:

3  $x_1$   $y_1$   $x_2$   $y_2$

que deve imprimir o valor do **somatório**, **mínimo** e **máximo** dos elementos da sub-matriz **(x1,y1,x2,y2)**.

Comandos deste tipo serão sempre os últimos da lista de **m** comandos. P. ex, se temos 8 comandos ( $m = 8$ ) e 3 comandos de impressão, tais comandos serão os 3 últimos desta sequência.

Caso haja algum erro, como p. ex., as dimensões das sub-matrizes estão fora das dimensões da matriz principal, entre outros, você deverá imprimir a palavra **erro** e terminar o programa.

Todas as restrições foram mencionadas na descrição.

## Exemplos

#	Entrada	Saída
1	5 5 7 1 1 2 4 4 5 1 1 1 3 4 2 2 2 1 4 4 2 1 1 1 4 3 3 3 1 1 3 4 3 1 2 4 4 3 0 0 4 4	66 2 10 63 2 10 83 0 10
2	5 5 2 1 0 0 6 2 5	erro

Obs: Notem que o comando de impressão imprimirá as saídas entre os comandos de entrada.

No terminal, teremos algo do tipo (as linhas em **negrito** correspondem às saídas):

```

5 5 7
1 1 2 4
4 5
1 1 1 3
4 2
2 2 1 4
4 2
1 1 1 4
3 3
3 1 1 3
4
66 2 10
3 1 2 4
4
63 2 10
3 0 0 4
4
83 0 10

```