

ALOCAÇÃO DINÂMICA E TIPOS ABSTRATOS DE DADOS

- Submissão com data e hora de entrega disponíveis na plataforma da disciplina.

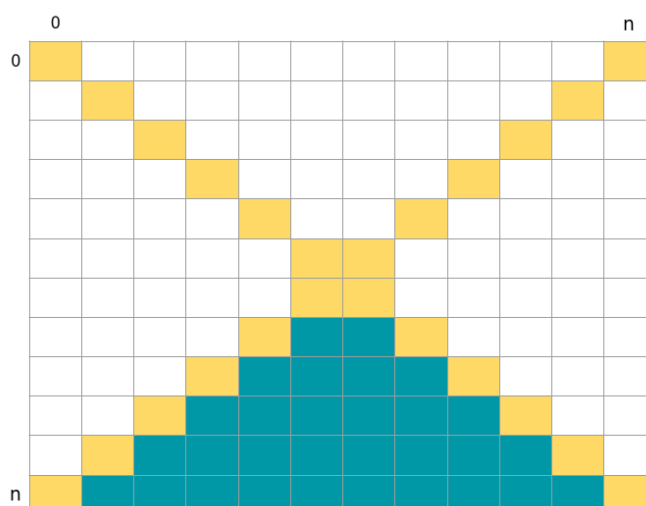
- Procedimento para a entrega:.

1. Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado.
2. Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas.
3. A avaliação considerará o tempo de execução e o percentual de respostas corretas.
4. Eventualmente, serão realizadas entrevistas sobre as práticas para complementar a avaliação.
5. Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada.
6. Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software.
7. Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota.
8. Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos.
9. Não serão considerados algoritmos parcialmente implementados.

- *Bom trabalho!*

Manipulação de Matrizes Dinâmicas

Crie um programa que leia um caractere maiúsculo, que indica uma operação que deve ser realizada em uma matriz de números reais $matriz[n][n]$, onde n corresponde as dimensões da matriz quadrada. As operações são: *S* para soma e *M* para média. Em seguida, calcule e mostre a soma ou a média considerando somente aqueles elementos que estão na área inferior da matriz, conforme ilustrado abaixo (área verde).



O código-fonte deve ser modularizado corretamente conforme os arquivos de protótipo fornecidos

Considerações

O código-fonte deve ser modularizado corretamente conforme os arquivos de protótipo fornecidos. Cada caso de teste deve ser resolvido em até 1 segundo!

- Não altere o nome dos arquivos.

- O arquivo `.zip` deve conter na sua raiz somente os arquivos-fonte.
- Há vários casos de teste. Você terá acesso (entrada e saída) de casos específicos para realizar os seus testes.

Especificação da Entrada e da saída

A primeira linha de entrada contém um único caractere maiúsculo: ‘*S*’ ou ‘*M*’, indicando a operação (Soma ou Média, respectivamente) que deverá ser realizada com os elementos da matriz. A próxima linha contém um inteiro n que representa as dimensões da matriz. Seguem **144** valores inteiros que compõem a matriz.

A saída é a impressão do resultado solicitado (a soma ou média), com 1 casa após o ponto decimal.

| Entrada | Saída |
|--|-------|
| S 12 1.0 330.0 -3.5 2.5 4.1 ... | 112.4 |

| Entrada | Saída |
|--|-------|
| M 12 1.0 330.0 -3.5 2.5 4.1 ... | 15.7 |

Diretivas de Compilação

```
$ gcc -c pratica.c -Wall
$ gcc -c matriz.c -Wall
$ gcc pratica.o matriz.o -o exe -lm
```

Avaliação de *leaks* de memória

Uma forma de avaliar se não há *leaks* de memória é usando a ferramenta valgrind. Um exemplo de uso é:

```
gcc -g -o exe *.c -Wall; valgrind --leak-check=yes -s ./exe < casoteste.in
```

Espera-se uma saída com o fim semelhante a:

```
==38409== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Para instalar no Linux, basta usar: `sudo apt install valgrind`.