

sqtpm

[186837]

[voltar](#)

Trabalho: 01-triangular

Linguagens: C

Data de abertura: 2016/08/23 12:00:00

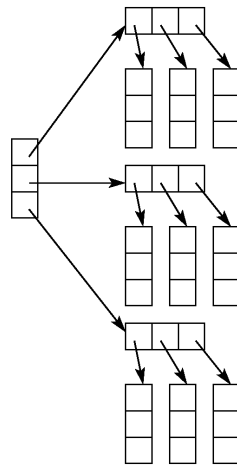
Data limite para envio: 2016/08/29 18:00:00 (encerrado)

Número máximo de envios: 25

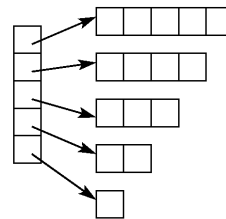
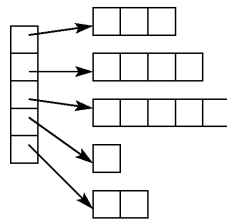
Casos-de-teste abertos: [casos-de-teste.tgz](#)

Em C, um array multidimensional é implementado colocando todos os elementos dele na memória sequencialmente, por linhas. Os elementos são acessados calculando um deslocamento a partir do primeiro elemento do array.

Outra forma de implementar um array com n dimensões é usando um array unidimensional (vetor) de apontadores para arrays com n-1 dimensões. Essa forma de implementação é usada por várias linguagens de programação. Dessa forma um array 3 x 3 x 3 seria implementado como na figura abaixo.

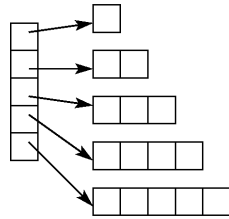


Um jagged array é um array multidimensional implementado usando apontadores e com elementos de tamanhos diferentes, por exemplo, matrizes triangulares e outras, como as que estão ilustradas abaixo.



sqtpm

[186837]

[voltar](#)

Escreva um programa para ler uma matriz triangular inferior de números fracionários fornecidos na entrada padrão, armazená-los dinamicamente na memória como um jagged array e calcular os valores normalizados. O programa deve usar a quantidade mínima de memória possível.

Entrada: A entrada consiste de um inteiro $n > 0$ especificando a ordem da matriz, seguido de $(n^2+n)/2$ fracionários contendo os elementos na matriz na seqüência:

```
[0,0]
[1,0] [1,1]
[2,0] [2,1] [2,2]
...
[n-1,0] [n-1,1] ... [n-1,n-1]
```

Saída: A saída deve ser iniciada por n linhas na forma de uma matriz triangular inferior. Cada valor z na matriz impressa na saída deve ser $z = (x-m)/d$, onde x é um elemento da matriz, m é a média e d é o desvio-padrão de todos os valores na matriz. Nenhuma entrada terá desvio-padrão igual a zero. Depois vem uma linha em branco e, na linha seguinte linha, a média e o desvio-padrão. Os valores devem ser impressos com 12 casas decimais. Há um espaço depois do último número em cada linha.

Exemplo:

Entrada:

```
5
2.3
1.4 8.2
3.1 2.0 7.7
5.3 6.1 4.4 1.2
5.5 6.1 3.0 2.3 4.9
```

Saída:

```
-0.892202112506
-1.307537578672 1.830552610141
-0.523015031469 -1.030647267895 1.599810684493
0.492249441383 0.861436522419 0.076913975216 -1.399834348932
0.584546211642 0.861436522419 -0.569163416599 -0.892202112506 0.307655900864

4.233333333333 2.166923061753
```

Uma pequena sugestão: Comece fazendo um programa que apenas lê a entrada e imprime a matriz na mesma ordem em que for lida, depois acrescente as demais funcionalidades.

Requisitos adicionais:

- Ao término da impressão da saída, o programa deve liberar a memória alocada dinamicamente