

## Lista de Exercícios

### Matrizes, Lista

1. Faça um programa que leia nome, idade e peso de várias pessoas, guardando estas informações em uma lista composta. Depois de sua execução mostre:
2. Quantas pessoas foram cadastradas
3. Uma lista com a(s) pessoa(s) mais pesada(s)
4. Uma lista com a(s) pessoa(s) mais leves(s)
5. O número de pessoas acima de 20 anos (mostrar também o nome e idade de cada uma)
6. O usuário define quantas pessoas serão cadastradas (Perguntar: Deseja continuar cadastrando?)

beecrowd | 1181

### Linha na Matriz

Por Neilor Tonin, URI  Brasil

**Timelimit: 1**

Neste problema você deve ler um número, indicando uma linha da matriz na qual uma operação deve ser realizada, um caractere maiúsculo, indicando a operação que será realizada, e todos os elementos de uma matriz  $M[12][12]$ . Em seguida, calcule e mostre a soma ou a média dos elementos que estão na área verde da matriz, conforme for o caso. A imagem abaixo ilustra o caso da entrada do valor 2 para a linha da matriz, demonstrando os elementos que deverão ser considerados na operação.



7. **Entrada**

**Entrada**

A primeira linha de entrada contém um número  $L$  ( $0 \leq L \leq 11$ ) indicando a linha que será considerada para operação. A segunda linha de entrada contém um único caractere Maiúsculo  $T$  ('S' ou 'M'), indicando a operação (Soma ou Média) que deverá ser realizada com os elementos da matriz. Seguem os 144 valores de ponto flutuante que compõem a matriz, sendo que a mesma é preenchida linha por linha, da linha 0 até a linha 11, sempre da esquerda para a direita.

**Saída**


Imprima o resultado solicitado (a soma ou média), com 1 casa após o ponto decimal.

Exemplo de Entrada	Exemplo de Saída
2 S 0.0 -3.5 2.5 4.1	12.6

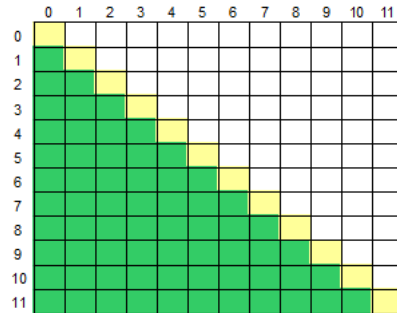
8. Adaptação do enunciado:
9. a. A ordem da matriz será lida do teclado
10. b. O preenchimento da matriz pode ser aleatório (usar random)
11. c. Imprimir a matriz antes e depois de efetuar o cálculo



# Abaixo da Diagonal Principal

Por Neilor Tonin, URI  Brasil**Timelimit: 1**

Leia um caractere maiúsculo, que indica uma operação que deve ser realizada e uma matriz  $M[12][12]$ . Em seguida, calcule e mostre a soma ou a média considerando somente aqueles elementos que estão abaixo da diagonal principal da matriz, conforme ilustrado abaixo (área verde).



## Entrada

### Entrada

A primeira linha de entrada contém um único caractere Maiúsculo **O** ('S' ou 'M'), indicando a operação (Soma ou Média) que deverá ser realizada com os elementos da matriz. Seguem os 144 valores de ponto flutuante que compõem a matriz.

## Saída

Imprima o resultado solicitado (a soma ou média), com 1 casa após o ponto decimal.

Exemplo de Entrada	Exemplo de Saída
S 1.0 0.0 -3.5 2.5 4.1 ...	12.6

Adaptação do enunciado:

- a. Ler o tamanho da matriz do teclado
- b. Deixar opção de preenchimento dos valores da matriz pelo teclado e também de forma randômica
- c. Imprimir a matriz

# Competição

Por Cristhian Bonilha, UTFPR 🇧🇷 Brazil

**Timelimit: 1**

A maioria dos programadores que chegam a escrever competições com exercícios de programação concordam em quatro características que toda competição deve alcançar. Embora nem todas sejam sempre alcançadas, quanto mais melhor. As características são as seguintes:

1. Ninguém resolveu todos os problemas.
2. Todo problema foi resolvido por pelo menos uma pessoa (não necessariamente a mesma).
3. Não há nenhum problema resolvido por todos.
4. Todos resolveram ao menos um problema (não necessariamente o mesmo).

Rafael organizou uma competição alguns dias atrás, e está preocupado com quantas dessas características ele conseguiu alcançar com a competição.

Dadas as informações sobre a competição, com o número de participantes, número de problemas, e qual participante resolveu quais problemas, descubra o número de características que foram alcançadas nesta competição.

## Entrada

Haverá diversos casos de teste. Cada caso de teste inicia com dois inteiros **N** e **M** ( $3 \leq N, M \leq 100$ ), indicando, respectivamente, o número de participantes e o número de problemas.

## Entrada

Haverá diversos casos de teste. Cada caso de teste inicia com dois inteiros **N** e **M** ( $3 \leq N, M \leq 100$ ), indicando, respectivamente, o número de participantes e o número de problemas.

Em seguida, haverá **N** linhas com **M** inteiros cada, onde o inteiro da linha **i** e coluna **j** é 1 caso o competidor **i** resolveu o problema **j**, ou 0 caso contrário.


O último caso de teste é indicado quando **N = M = 0**, o qual não deverá ser processado.

## Saída

Para cada caso de teste, imprima uma linha contendo um inteiro, representando quantas das características citadas foram alcançadas na competição.

Exemplo de Entrada	Exemplo de Saída
3 3 1 1 0 0 1 0 0 0 0 3 3 1 1 0 0 1 0 0 0 1 0 0	2 4

# Economia Brasileira

Por Carlos Andrade, UFMS  Brazil

**Timelimit: 2**

Ultimamente a economia brasileira tornou-se o assunto de todos os jornais. A população brasileira têm diferentes opiniões sobre o cenário econômico atual. Sua tarefa é fazer uma pesquisa para saber se a maioria da população está ou não satisfeita com o cenário econômico atual.

## Entrada

A primeira linha contém um inteiro  $Q$  ( $4 \leq Q \leq 233000$ ) representando o número de pessoas que participaram da pesquisa. A segunda linha contém  $Q$  inteiros  $V_i$  ( $0 \leq V_i \leq 1$ ,  $1 \leq i \leq Q$ ), representando a opinião do  $i$ -ésimo cidadão Brasileiro sobre o cenário econômico atual. Sendo "0" os que consideram o cenário atual satisfatório e "1" os que consideram não satisfatório.


## Saída

Seu programa deve imprimir "Y" caso a maioria da população esteja de acordo com o cenário econômico. Caso contrário imprima "N"

Exemplo de Entrada	Exemplo de Saída
5 1 1 1 1 1	N
4 1 1 0 0	N

beecrowd | 1471

# Mergulho

Maratona de Programação da SBC  Brasil

**Timelimit: 1**

O recente terremoto em Nlogônia não chegou a afetar muito as edificações da capital, principal epicentro do abalo. Mas os cientistas detectaram que o principal dique de contenção teve um dano significativo na sua parte subterrânea que, se não for consertado rapidamente, pode causar o seu desmoronamento, com a consequente inundação de toda a capital.

O conserto deve ser feito por mergulhadores, a uma grande profundidade, em condições extremamente difíceis e perigosas. Mas como é a sobrevivência da própria cidade que está em jogo, seus moradores acudiram em grande número como voluntários para essa perigosa missão.

Como é tradicional em missões perigosas, cada mergulhador recebeu no início do mergulho uma pequena placa com um número de identificação. Ao terminar o mergulho, os voluntários devolviam a placa de identificação, colocando-a em um repositório.

O dique voltou a ser seguro, mas aparentemente alguns voluntários não voltaram do mergulho. Você foi contratado para a penosa tarefa de, dadas as placas colocadas no repositório, determinar quais voluntários perderam a vida salvando a cidade.

## Entrada

A entrada contém vários casos de teste e termina com EOF. Cada caso de teste é composto de duas linhas. A primeira linha contém dois inteiros **N** e **R** ( $1 \leq R \leq N \leq 10^4$ ), indicando respectivamente o número de voluntários que mergulhou e o número de voluntários que retornou do mergulho. Os voluntários são identificados por números de 1 a N. A segunda linha da entrada contém R inteiros, indicando os voluntários que retornaram do mergulho (ao menos um voluntário retorna do mergulho).


## Saída

Seu programa deve produzir uma única linha para cada caso de teste, contendo os identificadores dos voluntários que não retornaram do mergulho, na ordem crescente de suas identificações. Deixe um espaço em branco após cada identificador (note que isto significa que deve haver um espaço em branco também após o último identificador). Se todos os voluntários retornaram do mergulho, imprima apenas o caractere '\*' (asterisco).

Exemplo de Entrada	Exemplo de Saída
5 3 3 1 5 6 6 6 1 3 2 5 4	2 4 *

beecrowd | 1961

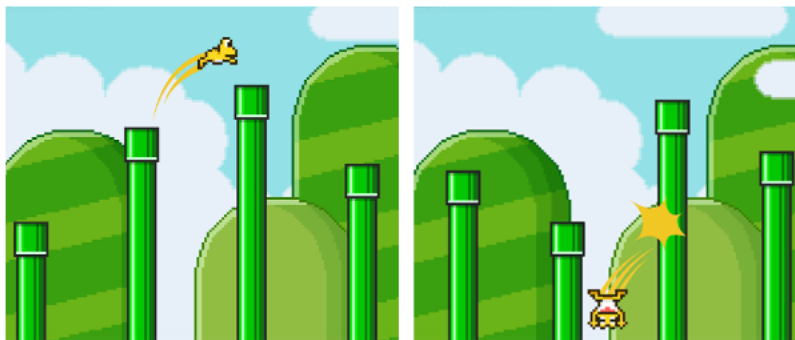
## Pula Sapo

Por M.C. Pinto, UNILA  Brazil

**Timelimit: 1**

Em cada fase do jogo do Pula Sapo você deve conduzir seu anfibio através de uma sequência de canos de alturas diferentes até chegar a salvo no cano mais à direita. Entretanto, o sapo só consegue sobreviver se a diferença de altura entre canos consecutivos for de, no máximo, a altura do pulo do sapo. Caso a altura do cano seguinte seja muito alta, o sapo bate no cano e cai. Se a altura do cano seguinte for muito baixa, o sapo não aguenta a queda. O sapo sempre começa em cima do cano mais à esquerda.

Neste jogo, a distância entre os canos é irrelevante, ou seja, o sapo sempre consegue alcançar o próximo cano com um pulo.



Você deve escrever um programa que, dadas as alturas dos canos e a altura do pulo do sapo, mostra se a fase do jogo pode ser vencida ou não.

### Entrada

A entrada é dada em duas linhas. A primeira tem dois inteiros positivos **P** e **N**, a altura do pulo do sapo e o número de canos ( $1 \leq P \leq 5$  e  $2 \leq N \leq 100$ ). A segunda linha tem **N** inteiros positivos que indicam as alturas dos canos ordenados da esquerda para a direita. Não há altura maior do que 10.

### Saída

A saída é dada em uma única linha. Se o sapo pode chegar no cano mais à direita, escreva "YOU WIN". Se o sapo não consegue, escreva "GAME OVER".

Exemplos de Entrada	Exemplos de Saída
5 10 1 3 6 9 7 2 4 5 8 3	YOU WIN
1 2 2 2	YOU WIN
1 2 1 3	GAME OVER

beecrowd | 1129

## Leitura Ótica

Maratona de Programação da SBC 🇧🇷 Brasil

**Timelimit: 1**

O professor João decidiu aplicar somente provas de múltipla escolha, para facilitar a correção. Em cada prova, cada questão terá cinco alternativas (A, B, C, D e E), e o professor vai distribuir uma folha de resposta para cada aluno. Ao final da prova, as folhas de resposta serão escaneadas e processadas digitalmente para se obter a nota de cada aluno. Inicialmente, ele pediu ajuda a um sobrinho, que sabe programar muito bem, para escrever um programa para extrair as alternativas marcadas pelos alunos nas folhas de resposta. O sobrinho escreveu uma boa parte do software, mas não pode terminá-lo, pois precisava treinar para a Maratona de Programação.

Durante o processamento, a prova é escaneada usando tons de cinza entre 0 (preto total) e 255 (branco total). Após detectar os cinco retângulos correspondentes a cada uma das alternativas, ele calcula a média dos tons de cinza de cada pixel, retornando um valor inteiro correspondente àquela alternativa. Se o quadrado foi preenchido corretamente o valor da média é zero (preto total). Se o quadrado foi deixado em branco o valor da média é 255 (branco total). Assim, idealmente, se os valores de cada quadrado de uma questão são (255, 0, 255, 255, 255), sabemos que o aluno marcou a alternativa B para essa questão. No entanto, como as folhas são processadas individualmente, o valor médio de nível de cinza para o quadrado totalmente preenchido não é necessariamente 0 (pode ser maior); da mesma forma, o valor para o quadrado não preenchido não é necessariamente 255 (pode ser menor). O prof. João determinou que os quadrados seriam divididos em duas classes: aqueles com média menor ou igual a 127 serão considerados pretos e aqueles com média maior a 127 serão considerados brancos.

Obviamente, nem todas as questões das folhas de resposta são marcadas de maneira correta. Pode acontecer de um aluno se enganar e marcar mais de uma alternativa na mesma questão, ou não marcar nenhuma

Obviamente, nem todas as questões das folhas de resposta são marcadas de maneira correta. Pode acontecer de um aluno se enganar e marcar mais de uma alternativa na mesma questão, ou não marcar nenhuma alternativa. Nesses casos, a resposta deve ser desconsiderada.

O professor João necessita agora de um voluntário para escrever um programa que, dados os valores dos cinco retângulos correspondentes às alternativas de uma questão determine qual a alternativa corretamente marcada, ou se a resposta à questão deve ser desconsiderada.

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um número inteiro **N** indicando o número de questões da folha de respostas ( $1 \leq N \leq 255$ ). Cada uma das **N** linhas seguintes descreve a resposta a uma questão e contém cinco números inteiros **A**, **B**, **C**, **D** e **E**, indicando os valores de nível de cinza médio para cada uma das alternativas da resposta ( $0 \leq A, B, C, D, E \leq 255$ ).

O ultimo caso de teste é seguido por uma linha que contém apenas um número zero.

Saída

Para cada caso de teste da entrada seu programa deve imprimir **N** linhas, cada linha correspondendo a uma questão. Se a resposta à questão foi corretamente preenchida na folha de resposta, a linha deve conter a alternativa marcada ('A', 'B', 'C', 'D' ou 'E'). Caso contrário, a linha deve conter o caractere '\*' (asterisco).

Exemplo de Entrada	Exemplo de Saída
3 0 255 255 255 255 255 255 255 255 0 255 255 127 255 255 4 200 200 200 0 200 200 1 200 200 1 1 2 3 4 5 255 5 200 130 205 0	A E C D * * B

beecrowd | 1164

Número Perfeito

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Na matemática, um número perfeito é um número inteiro para o qual a soma de todos os seus divisores positivos próprios (excluindo ele mesmo) é igual ao próprio número. Por exemplo o número 6 é perfeito, pois  $1+2+3$  é igual a 6. Sua tarefa é escrever um programa que imprima se um determinado número é perfeito ou não.

Entrada

A entrada contém vários casos de teste. A primeira linha da entrada contém um inteiro **N** ( $1 \leq N \leq 20$ ), indicando o número de casos de teste da entrada. Cada uma das **N** linhas seguintes contém um valor inteiro **X** ( $1 \leq X \leq 10^8$ ), que pode ser ou não, um número perfeito.

Saída

Para cada caso de teste de entrada, imprima a mensagem "**X** eh perfeito" ou "**X** nao eh perfeito", de acordo com a especificação fornecida.

Exemplo de Entrada	Exemplo de Saída
3 6 5 28	6 eh perfeito 5 nao eh perfeito 28 eh perfeito

10) Faça um programa que ajude um jogador da MEGASENA a criar palpites. O programa vai perguntar quantos jogos serão sorteados e vai gerar 6 números entre 1 e 60 para cada jogo, cadastrando tudo em uma única lista composta.

```
JOGA NA MEGA SENA
Quantos jogos você quer que eu sorteie? 4
===== SORTEANDO 4 JOGOS =====
Jogo 1: [4, 8, 10, 20, 26, 38]
Jogo 2: [14, 20, 23, 35, 38, 47]
Jogo 3: [2, 11, 15, 24, 43, 48]
Jogo 4: [6, 26, 28, 29, 30, 46]
===== < BOA SORTE! > =====
Process finished with exit code 0
```

12. Faça um programa que leia as seguintes informações dos alunos de POO: o nome, as 3 notas e calcule a média. Estas informações devem ser organizadas em uma lista composta (sendo que em cada posição da lista composta contém todas as informações de 1 aluno).

No final mostre um boletim contendo a média de cada aluno e permita que sejam consultadas todas as notas parciais de cada aluno. Esta consulta será feita pelo nome do aluno.