

Cursos de Verão - Instituto de Matemática e Estatística - USP**Introdução à Programação****5ª Lista de Exercícios - Entrega 07-02-2013****Instruções:**

1. Essa 5ª Lista de Exercícios possui 28 problemas que estão no URI Online Judge¹ onde cada um vale 0.5 (totalizando 14 pontos).
2. Não é obrigatório usar o URI Online Judge, pois o código das questões ainda deve ser colocado no Paca e o código colocado lá é que vai ser avaliado.
3. Entretanto, recomenda-se que use o UOJ para verificação automática (e quase que imediata) das suas soluções se estão corretas ou não.
4. Esta última lista poderá ser feita em grupo com no máximo 3 pessoas (a Maratona de Programação é em times de 3 integrantes), onde apenas uma delas deve colocar os códigos *.c no Paca seguindo o padrão definido a seguir:
 - O nome do arquivo deve ser o código da questão no URI;
 - Todos os arquivos .c devem ter no cabeçalho o nome de todos os integrantes do grupo conforme mostrado o padrão exemplo seguir:

```
1  /*
2  *  Introducao a Programacao – VERA0 2013 – IME – USP
3  *  Prof. Neuton de Oliveira Braga Jr
4  *  5 Lista de Exercicios
5  *
6  *  URI Online Judge – www.urionlinejudge.com.br
7  *  Problema 9999
8  *
9  *  Aluno 1: Fulano de Tal
10 *  Aluno 2: Beltrano de Tal
11 *  Aluno 3: Cicrano de Tal
12 *
13 *  Turma: Madrugada
14 */
15
16 #include <stdio.h>
17
18 /* Inclua a biblioteca math.h e outras, se forem usa-las */
19
20 /* Declare funcoes se for necessario */
21
22 int main() {
23     printf("Esse_eh_somente_um_codigo_modelo_para_lista_5.\n");
24
25     return 0;
26 }
```

5. Leitura e Interpretação é fundamental na resolução dos problemas. Leia com atenção todas as questões e responda elas de acordo com o que é pedido.

¹<http://www.urionlinejudge.com.br/>

6. Se um problema estiver muito complicado de fazer, tente primeiro pensar e escrever no papel o algoritmo com a lógica de programação que você identificou e somente depois tente escrever o código C.
7. Resolva as questões utilizando boas práticas de programação em C (identação, nomenclaturas, etc).
8. Nas questões que envolvem elaboração de programas, utilize a linguagem **C** no padrão **ANSI** (mesmo que você submeta no UOJ como C++).
9. Lembre-se tudo que é entrada deve ser tratada usando alguma função de entrada como, por exemplo, scanf. A saída é tudo que é mostrado com alguma função para saída de dados como, por exemplo, printf.
10. Qualquer dúvida que estiver com entrada, saída ou outras, coloque no Paga (sem o código) ou mande um e-mail para o Professor e para o Monitor da disciplina com seu código em anexo.
11. Os problemas estão mais ou menos em ordem crescente de dificuldade e abrangem assuntos vistos desde o início do curso. Assim, procure fazer os mais fáceis para depois passar para os mais difíceis.
12. Não seguir qualquer uma das instruções poderá implicar penalizações da nota.

URI Online Judge | 1078

Tabuada

*Adaptado por Neilor Tonin, URI  Brasil***Timelimit: 1**

Leia 1 valor inteiro N ($2 < N < 1000$). A seguir, mostre a tabuada de N :

$1 \times N = N$ $2 \times N = 2N$... $10 \times N = 10N$

Entrada

A entrada contém um valor inteiro N ($2 < N < 1000$).

Saída

Imprima a tabuada de N , conforme o exemplo fornecido.

Exemplo de Entrada	Exemplo de Saída
140	$1 \times 140 = 140$ $2 \times 140 = 280$ $3 \times 140 = 420$ $4 \times 140 = 560$ $5 \times 140 = 700$ $6 \times 140 = 840$ $7 \times 140 = 980$ $8 \times 140 = 1120$ $9 \times 140 = 1260$ $10 \times 140 = 1400$

URI Online Judge | 1164

Número Perfeito

*Adaptado por Neilor Tonin, URI  Brasil***Timelimit: 1**

Na matemática, um número perfeito é aquele que é igual à soma dos seus divisores. 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 eh perfeito
6	5 nao eh perfeito
5	28 eh perfeito
28	

URI Online Judge | 1037

Intervalo

*Adaptado por Neilor Tonin, URI  Brasil***Timelimit: 1**

O arquivo de entrada contém um valor real qualquer. O programa deve apresentar uma mensagem dizendo em qual dos seguintes intervalos: $[0,25]$ $(25,50]$, $(50,75]$, $(75,100]$. Se o valor for menor do que 0 ou maior do que 100 deve ser apresentada uma mensagem "Fora de intervalo".

Por exemplo:

$[0,25]$ indica valores entre 0 e 25.0000, inclusive eles.

$(25,50]$ indica valores maiores que 25 Ex: 25.00001 até o valor 50.0000000

Entrada

O arquivo de entrada contém um número com ponto flutuante qualquer.

Saída

A saída deve ser uma mensagem conforme exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
25.02	Intervalo (25,50]

URI Online Judge | 1042

Sort Simples

*Adaptado por Neilor Tonin, URI  Brasil***Timelimit: 1**

Leia 3 valores inteiros e ordene-os em ordem crescente. No final, mostre os valores em ordem crescente, uma linha em branco e em seguida, os valores na sequência como foram lidos.

Entrada


A entrada contém três números inteiros.

Saída

Imprima a saída conforme foi especificado.

Exemplo de Entrada	Exemplo de Saída
7 21 -14	-14 7 21 7 21 -14

Crescimento Populacional

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Mariazinha quer resolver um problema interessante. Dadas as informações de população e a taxa de crescimento de duas cidades quaisquer (A e B), ela gostaria de saber quantos anos levará para que a cidade menor (sempre é a cidade A) ultrapasse a cidade B em população. Claro que ela quer saber apenas para as cidades cuja taxa de crescimento da cidade A é maior do que a taxa de crescimento da cidade B, portanto, previamente já separou para você apenas os casos de teste que tem a taxa de crescimento maior para a cidade A. Sua tarefa é construir um programa que apresente o tempo em anos para cada caso de teste.

Porém, em alguns casos o tempo pode ser muito grande, e Mariazinha não se interessa em saber exatamente o tempo para estes casos. Basta que você informe, nesta situação, a mensagem "Mais de 1 seculo".

Entrada

A primeira linha da entrada contém um único inteiro **T**, indicando o número de casos de teste ($1 \leq T \leq 3000$). Cada caso de teste contém 4 números: dois inteiros **P1** e **P2** ($1 \leq P1 \leq 1000000$, $1 \leq P2 \leq 1000000$, $P1 \leq P2$) indicando respectivamente a população de A e B, e dois valores **G1** e **G2** ($0.1 \leq G1 \leq 10.0$, $0.1 \leq G2 \leq 10.0$, $G2 \leq G1$) com um dígito após o ponto decimal cada, indicando respectivamente o crescimento populacional de A e B.

Saída

Imprima, para cada caso de teste, quantos anos levará para que a cidade A ultrapasse a cidade B em número de habitantes. Obs.: se o tempo for mais do que 100 anos o programa deve apresentar a mensagem: Mais de 1 seculo. Neste caso, acredito que seja melhor interromper o programa imediatamente após passar de 100 anos, caso contrário você poderá receber como resposta da submissão deste problema "Time Limit Exceeded".

Exemplo de Entrada	Exemplo de Saída
5	16 anos.
90000 120000 5.5 3.5	12 anos.
56700 72000 5.2 3.0	Mais de 1 seculo.
123 2000 3.0 2.0	10 anos.
100000 110000 1.5 0.5	100 anos.
62422 484317 3.1 1.0	

URI Online Judge | 1036

Fórmula de Báscara

*Adaptado por Neilor Tonin, URI  Brasil***Timelimit: 1**

Leia 3 valores inteiros e efetue o cálculo das raízes da equação de báscara. Se não foi possível calcular as raízes, mostre a mensagem correspondente *"Impossível calcular"*, caso haja uma divisão por 0 ou raiz de numero negativo.

Entrada

Leia três numeros inteiros A, B and C.

Saída

Se não houver possibilidade de calcular as raízes, apresente a mensagem "Impossível calcular". Caso contrário, imprima o resultado das raízes com 5 dígitos após o ponto, com uma mensagem correspondente conforme exemplo abaixo. Imprima sempre o final de linha após cada mensagem.

Exemplo de Entrada	Exemplo de Saída
2 10 -5	R1 = 0.45804 R2 = -5.45804

URI Online Judge | 1021

Notas e Moedas

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

Leia um valor de ponto flutuante com duas casas decimais. Este valor representa um valor monetário. A seguir, calcule o menor número de notas e moedas possíveis no qual o valor pode ser decomposto. As notas consideradas são de 100, 50, 20, 10, 5, 2. As moedas possíveis são de 1, 0.50, 0.25, 0.10, 0.05 e 0.01. A seguir mostre a relação de notas necessárias.

Entrada

O arquivo de entrada contém um valor de ponto flutuante **N** ($0 \leq N \leq 1000000.00$).


Saída

Imprima a quantidade mínima de notas e moedas necessárias para trocar o valor inicial, conforme exemplo fornecido.

Exemplo de Entrada	Exemplo de Saída
576.73	NOTAS: 5 nota(s) de R\$ 100.00 1 nota(s) de R\$ 50.00 1 nota(s) de R\$ 20.00 0 nota(s) de R\$ 10.00 1 nota(s) de R\$ 5.00 0 nota(s) de R\$ 2.00 MOEDAS: 1 moeda(s) de R\$ 1.00 1 moeda(s) de R\$ 0.50 0 moeda(s) de R\$ 0.25 2 moeda(s) de R\$ 0.10 0 moeda(s) de R\$ 0.05 3 moeda(s) de R\$ 0.01

URI Online Judge | 1172

Substituição em Vetor I

*Adaptado por Neilor Tonin, URI  Brasil***Timelimit: 1**

Faça um programa que leia um vetor $X[10]$. Substitua a seguir, todos os valores nulos e negativos do vetor X por 1. Em seguida mostre o vetor X .

Entrada

A entrada contém 10 valores inteiros, podendo ser positivos ou negativos.

Saída

Para cada posição do vetor, escreva " $X[i] = x$ ", onde i é a posição do vetor e x é o valor armazenado naquela posição.

Exemplo de Entrada	Exemplo de Saída
0	$X[0] = 1$
-5	$X[1] = 1$
63	$X[2] = 63$
0	$X[3] = 1$
...	...

Agradecimentos a Cassio F.

URI Online Judge | 1140

Flores Florescem da França

por Ines Kereki  *Uruguai***Timelimit: 1**

Fiona sempre amou poesia, e recentemente descobriu uma forma poética fascinante. *Tautogramas* são um caso especial de aliteração, que é a ocorrência da mesma letra no início de palavras adjacentes. Em particular, uma sentença é um tautograma se todas suas palavras começam com a mesma letra.

Por exemplo, as seguintes sentenças são tautogramas:

- Flowers Flourish from France
- Sam Simmonds speaks softly
- Peter pIckEd pePPers
- truly tautograms triumph

Fiona quer deslumbrar seu namorado com uma carta romântica repleta desse tipo de sentenças. Por favor, ajude Fiona a verificar se cada sentença que ela escreveu é um tautograma ou não.

Entrada

Cada caso de teste é dado em uma única linha que contém uma frase. Uma frase consiste de uma sequência de no máximo 50 palavras separadas por espaços simples. Uma palavra é uma sequência de no máximo 20 letras do alfabeto inglês (maiúsculas e minúsculas). Uma palavra contém ao menos uma letra e uma sentença contém ao menos uma palavra.

O último caso de teste é seguido de uma linha contendo apenas o caractere '*' (asterisco).


Saída

Para cada caso de teste, imprima uma única linha contendo o caractere 'Y' se a sentença é um tautograma, ou um 'N' caso contrário.

Exemplo de Entrada	Exemplo de Saída
Flowers Flourish from France	Y
Sam Simmonds speaks softly	Y
Peter pIckEd pePPers	Y
truly tautograms triumph	Y
this is NOT a tautogram	N
*	

URI Online Judge | 1192

O jogo Matemático de Paula

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

Paula simplesmente adora matemática. Seu maior passatempo é ficar inventando jogos ou atividades que a envolvam para brincar com seus amiguinhos. Obviamente, nem todos eles não são tão apaixonados assim por matemática e têm muita dificuldade para resolver as brincadeiras propostas por ela. Agora Paula inventou um pequeno passatempo que envolve 3 caracteres: um dígito numérico, uma letra e outro dígito numérico.

Se a letra for maiúscula, deve-se subtrair o primeiro dígito do segundo. Se a letra for minúscula, deve-se somar ambos os dígitos e se os DÍGITOS forem iguais, deve-se desconsiderar a letra e mostrar o produto entre os dois dígitos. Ela pediu para seu amigo Marcelo, que é bom em programação, para criar um programa para que encontre a solução para cada uma das sequências que Paula lhe apresentar.

Entrada

A entrada contém vários casos de teste. A primeira linha da entrada contém um inteiro N , indicando o número de casos de teste que virão a seguir. Cada caso de teste é uma sequência de três caracteres criada por Paula. Esta sequência contém na primeira posição um caractere de '0' a '9', na segunda posição uma letra maiúscula ou minúscula do alfabeto e na terceira posição outro caractere de '0' a '9'.

Saída

Para cada caso de teste, deve ser impressa uma linha com a solução da sequência proposta por Paula.

Exemplo de Entrada	Exemplo de Saída
5	1
4A5	9
3A3	6
4f2	2
2G4	-6
7Z1	

URI Online Judge | 1177

Preenchimento de Vetor II

*Adaptado por Neilor Tonin, URI  Brasil***Timelimit: 1**

Faça um programa que leia um valor **T** e preencha um vetor **N**[1000] com a sequência de valores de 0 até **T**-1 repetidas vezes, conforme exemplo abaixo. Imprima o vetor **N**.

Entrada

A entrada contém um valor inteiro **T** ($2 \leq T \leq 50$).

Saída


Para cada posição do vetor, escreva "**N**[*i*] = **x**", onde *i* é a posição do vetor e **x** é o valor armazenado naquela posição.

Exemplo de Entrada	Exemplo de Saída
3	N[0] = 0 N[1] = 1 N[2] = 2 N[3] = 0 N[4] = 1 N[5] = 2 N[6] = 0 N[7] = 1 N[8] = 2 ...

Agradecimentos a Cassio F.

URI Online Judge | 1179

Preenchimento de Vetor IV

*Adaptado por Neilor Tonin, URI  Brasil***Timelimit: 1**

Neste problema você deverá ler 15 valores colocá-los em 2 vetores conforme estes valores forem pares ou ímpares. Só que o tamanho de cada um dos dois vetores é de 5 posições. Então, cada vez que um dos dois vetores encher, você deverá imprimir todo o vetor e utilizá-lo novamente para os próximos números que forem lidos. Terminada a leitura, deve-se imprimir o conteúdo que restou em cada um dos dois vetores, imprimindo primeiro os valores do vetor ímpar. Cada vetor pode ser preenchido tantas vezes quantas for necessário.

Entrada

A entrada contém 15 números inteiros.


Saída

Imprima a saída conforme o exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
1	par[0] = 4
3	par[1] = -4
4	par[2] = 2
-4	par[3] = 8
2	par[4] = 2
3	impar[0] = 1
8	impar[1] = 3
2	impar[2] = 3
5	impar[3] = 5
-7	impar[4] = -7
54	impar[0] = 789
76	impar[1] = 23
789	par[0] = 54
23	par[1] = 76
98	par[2] = 98

URI Online Judge | 1024

Criptografia

Por Neilor Tonin, URI  Brasil

Timelimit: 1

Você foi solicitado para construir um programa simples de criptografia. Este programa deve possibilitar enviar mensagens codificadas sem que alguém consiga lê-las. O processo é muito simples. São feitas três passadas em todo o texto.

Na primeira passada, números e caracteres especiais não devem ser alterados. Cada um dos caracteres restantes deve ser deslocado 3 posições para a direita, segundo a tabela ASCII: letra 'a' deve virar letra 'd', letra 'y' deve virar caractere '|' e assim sucessivamente. Na segunda passada, a linha deverá ser invertida. Na terceira e última passada, os caracteres a partir da metade em diante (truncada) devem ser deslocados uma posição para a esquerda na tabela ASCII. Neste caso, 'b' vira 'a' e 'a' vira `.`.

Por exemplo, se a entrada for "Texto #3", o primeiro processamento sobre esta entrada deverá produzir "Wh{wr #3". O resultado do segundo processamento inverte os caracteres e produz "3# rw{hW". Por último, com o deslocamento dos caracteres da metade em diante, o resultado final deve ser "3# rvzgV".

Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém um inteiro **N** ($1 \leq N \leq 1 \cdot 10^4$), indicando a quantidade de linhas que o problema deve tratar. As **N** linhas contém cada uma delas **M** ($1 \leq M \leq 1 \cdot 10^3$) caracteres.

Saída

Para cada entrada, deve-se apresentar a mensagem criptografada.

Exemplo de Entrada	Exemplo de Saída
4 Texto #3 abcABC1 vxpdylY .ph vv.xwfxo.fd	3# rvzgV 1FECedc ks. \n{frzx gi.r{hyz-xx

Concurso de Contos

Maratona de Programação da SBC 2012  Brasil

Timelimit: 1

Machado gosta muito de escrever. Já escreveu muitos contos, resenhas, relatos de viagens que fez, além de um pequeno romance. Agora Machado quer participar de um concurso de contos, que tem regras muito rígidas sobre o formato de submissão do conto. As regras do concurso especificam o número máximo de caracteres por linha, o número máximo de linhas por página, além de limitar o número total de páginas. Adicionalmente, cada palavra deve ser escrita integralmente em uma linha (ou seja, a palavra não pode ser separada silabicamente em duas linhas). Machado quer escrever um conto com o maior número de palavras possível, dentro das regras do concurso, e precisa de sua ajuda. Dados o número máximo de caracteres por linha, o número máximo de linhas por página, e as palavras do conto que Machado está escrevendo, ele quer saber o número mínimo de páginas que seu conto utilizaria seguindo as regras do concurso.

Entrada

A primeira linha de um caso de teste contém três inteiros **N** ($2 \leq N \leq 1000$), **L** ($1 \leq L \leq 30$) e **C** ($1 \leq C \leq 70$), que indicam, respectivamente, o número de palavras do conto de Machado, o número máximo de linhas por página e o número máximo de caracteres por linha. O conto de Machado é inovador e não contém nenhum caractere além de letras maiúsculas e minúsculas e espaços em branco, sem letras acentuadas e sem cedilha. A segunda linha contém o conto de Machado, composto de **N** palavras ($1 \leq \text{comprimento de cada palavra} \leq C$) separadas por espaços em branco; há espaço em branco somente entre duas palavras, e entre duas palavras há exatamente um espaço em branco.

Saída

Para cada caso de teste imprima uma única linha, contendo um único número inteiro, indicando o número mínimo de páginas que o conto de Machado ocupa, considerando as regras do concurso.

Exemplo de Entrada	Exemplo de Saída
14 4 20 Se mana Piedade tem casado com Quincas Borba apenas me daria uma esperanca colateral	2 1 3 3
16 3 30 No dia seguinte entrou a dizer de mim nomes feios e acabou alcunhando me Dom Casmurro	
5 2 2 a de i de o	
5 2 2 a e i o u	

URI Online Judge | 1235

De Dentro para Fora

Por TopCoder*  EUA

Timelimit: 1

A sua impressora foi infectada por um vírus e está imprimindo de forma incorreta. Depois de olhar para várias páginas impressas por um tempo, você percebe que ele está imprimindo cada linha de dentro para fora. Em outras palavras, a metade esquerda de cada linha está sendo impressa a partir do meio da página até a margem esquerda. Do mesmo modo, a metade direita de cada linha está sendo impressa à partir da margem direita e prosseguindo em direção ao centro da página.

Por exemplo a linha:

THIS LINE IS GIBBERISH

está sendo impressa como:

I ENIL SIHTHSIREBBIG S

Da mesma foma, a linha " MANGOS " está sendo impressa incorretamente como "NAM SOG". Sua tarefa é desembaralhar (decifrar) a string a partir da forma como ela foi impressa para a sua forma original. Você pode assumir que cada linha conterà um número par de caracteres.

Entrada

A entrada contém vários casos de teste. A primeira linha de entrada contém um inteiro **N** que indica a quantidade de casos de teste. Seguem **N** linhas, cada uma com uma frase com no mínimo 2 e no máximo 100 caracteres de letras maiúsculas e espaços que deverá ser desembaralhada (decifrada) à partir da forma impressa para a sua forma original, conforme especificação acima.

Saída

Para cada linha de entrada deverá ser impressa uma linha de saída com a frase decifrada, conforme a especificação acima.

Exemplo de Entrada	Exemplo de Saída
5	THIS LINE IS GIBBERISH
I ENIL SIHTHSIREBBIG S	LEVELKAYAK
LEVELKAYAK	HAPPY HOLIDAYS
H YPPAHSYADILO	MLKJIHGFEDCBAZYXWUTSRQPON
ABCDEFGHIJKLMNOPQRSTUVWXYZ	THREE FRENCH HENS TWO TURTLEDOVES AND A
RUT OWT SNEH HCNERF EERHTEGDIRTRAP A	PARTRIDGE
DNA SEVODELT	


* Este problema é de autoria do TopCoder (www.topcoder.com/tc) e foi adaptado por Neilor para utilização (autorizada) no URI OJ.

* A reprodução não autorizada deste problema sem o consentimento por escrito de TopCoder, Inc. é

estritamente proibida.

URI Online Judge | 1263

Aliteração

Por TopCoder*  EUA

Timelimit: 1

Uma aliteração ocorre quando duas ou mais palavras consecutivas de um texto possuem a mesma letra inicial (ignorando maiúsculas e minúsculas). Sua tarefa é desenvolver um programa que identifique, a partir de uma sequência de palavras, o número de aliterações que essa sequência possui.

Entrada

A entrada contém diversos casos de testes. Cada caso é expresso como um texto em uma única linha, contendo de 1 a 100 palavras separadas por um único espaço, cada palavra tendo de 1 a 50 letras minúsculas ou maiúsculas ('A'-'Z','a'-'z'). A entrada termina em EOF.

Saída

Para cada caso de teste imprima o número de aliterações existentes no texto informado, conforme exemplos abaixo.


Exemplo de Entrada	Exemplo de Saída
He has four fanatic fantastic fans	2
There may be no alliteration in a sequence	0 2
Round the rugged rock the ragged rascal ran	3
area artic Soul Silly subway ant artic none	

* Este problema é de autoria do TopCoder (www.topcoder.com/tc) e foi adaptado por bitfreeze para utilização (autorizada) no URI OJ.

* A reprodução não autorizada deste problema sem o consentimento por escrito de TopCoder, Inc. é estritamente proibida.

URI Online Judge | 1182

Coluna na Matriz

Por Neilor Tonin, URI  Brasil

Timelimit: 1

Neste problema você deve ler um número que indica uma coluna de uma 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 5 para a coluna da matriz, demonstrando os elementos que deverão ser considerados na operação.

	0	1	2	3	4	5	6	7	8	9	10	11
0												
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												

Entrada

A primeira linha de entrada contém um número C ($0 \leq C \leq 11$) indicando a coluna 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.

Saída

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

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

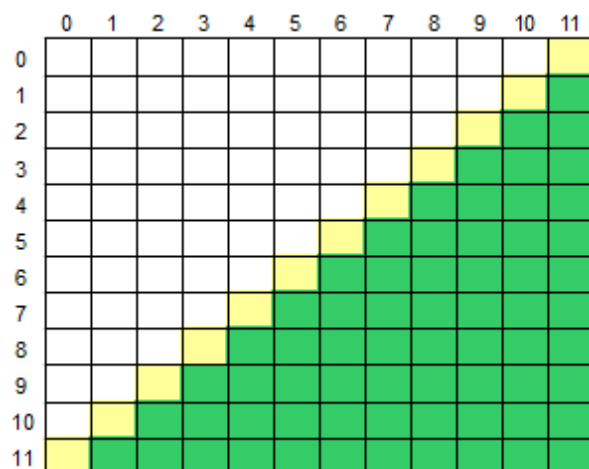
URI Online Judge | 1186

Abaixo da Diagonal Secundária

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 Secundária da matriz, conforme ilustrado abaixo (área verde).



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.

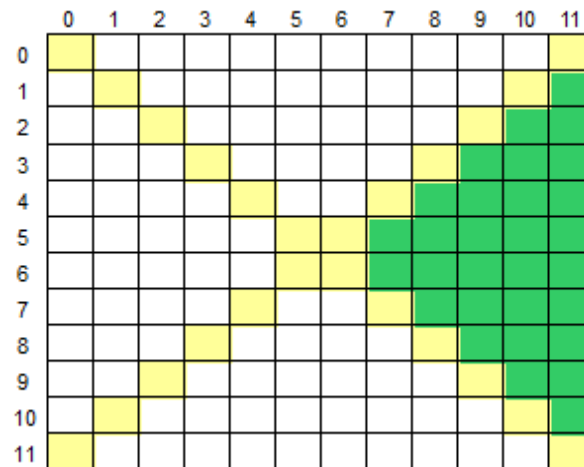
Exemplo de Entrada	Exemplo de Saída
S 1.0 0.0 -3.5 2.5 4.1 ...	12.6

URI Online Judge | 1190

Área Direita

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 na área direita da matriz, conforme ilustrado abaixo (área verde).



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 1.0 330.0 -3.5 2.5 4.1 ...	111.4

URI Online Judge | 1089

Loop Musical

Por Ricardo Anido  Brasil**Timelimit: 1**

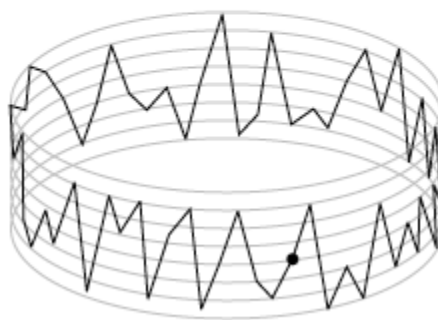
Um *loop musical* é um trecho de música que foi composto para repetir continuamente (ou seja, o trecho inicia novamente toda vez que chega ao final), sem que se note descontinuidade. Loops são muito usados na sonorização de jogos, especialmente jogos casuais pela internet.

Loops podem ser digitalizados por exemplo utilizando PCM. PCM, do inglês Pulse Code Modulation, é uma técnica para representação de sinais analógicos, muito utilizada em áudio digital. Nessa técnica, a magnitude do sinal é amostrada a intervalos regulares de tempo, e os valores amostrados são armazenados em sequência. Para reproduzir a forma de onda amostrada, o processo é invertido (demodulação).

Fernandinha trabalha para uma empresa que desenvolve jogos e compôs um bonito loopmusical, codificando-o em PCM. Analisando a forma de onda do seu loop em um software de edição de áudio, Fernandinha ficou curiosa ao notar a quantidade de “picos” existentes. Um pico em uma forma de onda é um valor de uma amostra que representa um máximo ou mínimo local, ou seja, um ponto de inflexão da forma de onda. A figura abaixo ilustra (a) um exemplo de forma de onda e (b) o loop formado com essa forma de onda, contendo 48 picos.



(a) Uma forma de onda



(b) A mesma forma de onda como um loop

Fernandinha é uma amiga muito querida e pediu sua ajuda para determinar quantos picos existem no seu loop musical.

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um inteiro N ,

representando o número de amostras no loop musical de Fernandinha ($2 \leq N \leq 10^4$). A segunda linha contém N inteiros H_i separados por espaços, representando a sequência de magnitudes das amostras ($-10^4 \leq H_i \leq 10^4$ for $1 \leq i \leq N$, $H_1 \neq H_N$ and $H_i \neq H_{i+1}$ for $1 \leq i < N$). Note que H_1 segue H_N quando o loop é reproduzido.

O final da entrada é indicado por uma linha que contém apenas o número zero.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha, contendo apenas um inteiro, o número de picos existentes no loop musical de Fernandinha.

Exemplo de Entrada	Exemplo de Saída
2	2
1 -3	2
6	4
40 0 -41 0 41 42	
4	
300 450 449 450	
0	

Maratona de Programação da SBC 2008.

URI Online Judge | 1138

Contagem de Dígitos

por Pablo Ariel Heiber  *Argentina***Timelimit: 2**

Diana escreverá uma lista com todos os inteiros positivos entre **A** e **B**, inclusive, na base decimal e sem zeros à esquerda. Ela quer saber quantas vezes cada um dos dígitos irá ser usado.

Entrada

Cada caso de teste é dado em uma única linha que contém dois inteiros **A** e **B** ($1 \leq A \leq B \leq 10^8$). O último caso de teste é seguido por uma linha contendo dois zeros.

Saída

Para cada caso de teste, imprima uma única linha com 10 inteiros representando o número de vezes que cada dígito é usado ao escrever todos os inteiros entre **A** e **B**, inclusive, na base decimal e sem zeros à esquerda. Escreva a contagem de cada dígito em ordem crescente do 0 até o 9.

Exemplo de Entrada	Exemplo de Saída
1 9	0 1 1 1 1 1 1 1 1 1
12 321	61 169 163 83 61 61 61 61 61 61
5987 6123	134 58 28 24 23 36 147 24 27 47
12345678 12345679	0 2 2 2 2 2 2 2 1 1
0 0	

KiloMan

Timelimit: 1

Entrada

Saída

Exemplo de Entrada	Exemplo de Saída
4 9 1 3 2 3 3 1 2 2 1 JJSSSJSSJ 49 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 7 7 SSSSSSSSSSSSSSJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ 4 1 2 2 1 SJJS 1 1 J	4 49 2 0

* Este problema é de autoria do TopCoder (www.topcoder.com/tc) e foi adaptado por Julio B. para utilização (autorizada) no URI OJ.

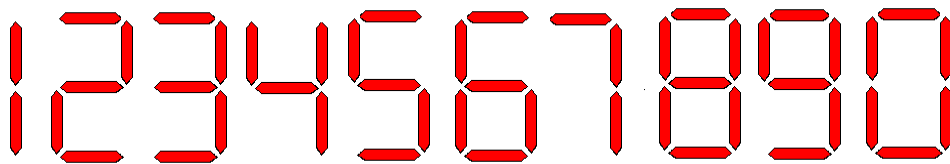
* A reprodução não autorizada deste problema sem o consentimento por escrito de TopCoder, Inc. é estritamente proibida.

URI Online Judge | 1168

LED

*Autor Desconhecido***Timelimit: 1**

João quer montar um painel de leds contendo diversos números. Ele não possui muitos leds, e não tem certeza se conseguirá montar o número desejado. Considerando a configuração dos leds dos números abaixo, faça um algoritmo que ajude João a descobrir a quantidade de leds necessário para montar o valor.



Entrada

A entrada contém um inteiro **N**, ($1 \leq N \leq 1000$) correspondente ao número de casos de teste, seguido de **N** linhas, cada linha contendo um número ($1 \leq V \leq 10^{100}$) correspondente ao valor que João quer montar com os leds.

Saída

Para cada caso de teste, imprima uma linha contendo o número de leds que João precisa para montar o valor desejado, seguido da palavra "leds".

Exemplo de Entrada	Exemplo de Saída
3	27 leds
115380	29 leds
2819311	25 leds
23456	

Agradecimentos a Cassio F.

URI Online Judge | 1278

Justificador II

Por Bruno Santos  Brasil

Timelimit: 1

Nós temos alguns textos e queremos formatá-los e justificá-los à direita, ou seja, alinhar suas linhas à margem direita de cada um. Crie um programa que, após ler um texto, reimprima esse texto com apenas um espaço entre as palavras e suas linhas justificadas à direita em todo o texto.

Entrada

A entrada contém diversos casos de teste. A primeira linha de cada caso de teste contém um inteiro **N** ($1 \leq N \leq 100$) que indica o número de linhas de texto que virão a seguir. Cada uma destas **N** linhas de texto contém de 1 a 50 letras maiúsculas ('A'-'Z') ou espaços (' '). Todas as linhas de texto contém no mínimo uma letra. Poderá haver mais de um espaço entre as palavras. É também possível haver espaços no início e no final da linha. O fim da entrada é indicado por **N** = 0.

Saída

Para cada caso de teste imprima o texto com apenas um espaço entre as palavras, e inserindo tantos espaços quanto forem necessários à esquerda de cada linha do texto, para que elas apareçam alinhadas à margem direita daquele texto, e na mesma ordem da entrada. Deixe uma linha em branco entre os casos de testes. Não imprima espaços no final de cada linha, nem espaços desnecessários à esquerda, de modo que pelo menos uma das linhas impressa em cada texto inicie com uma letra.

Exemplo de Entrada	Exemplo de Saída
3 ROMEO AND JULIET WERE IN LOVE	ROMEO AND JULIET WERE IN LOVE
4 WHO ELSE LOVES STAIRS	WHO ELSE LOVES STAIRS
3 A TEXT CAN BE JUSTIFIED ON BOTH SIDES OR JUST TO THE RIGHT	A TEXT CAN BE JUSTIFIED ON BOTH SIDES OR JUST TO THE RIGHT
0	

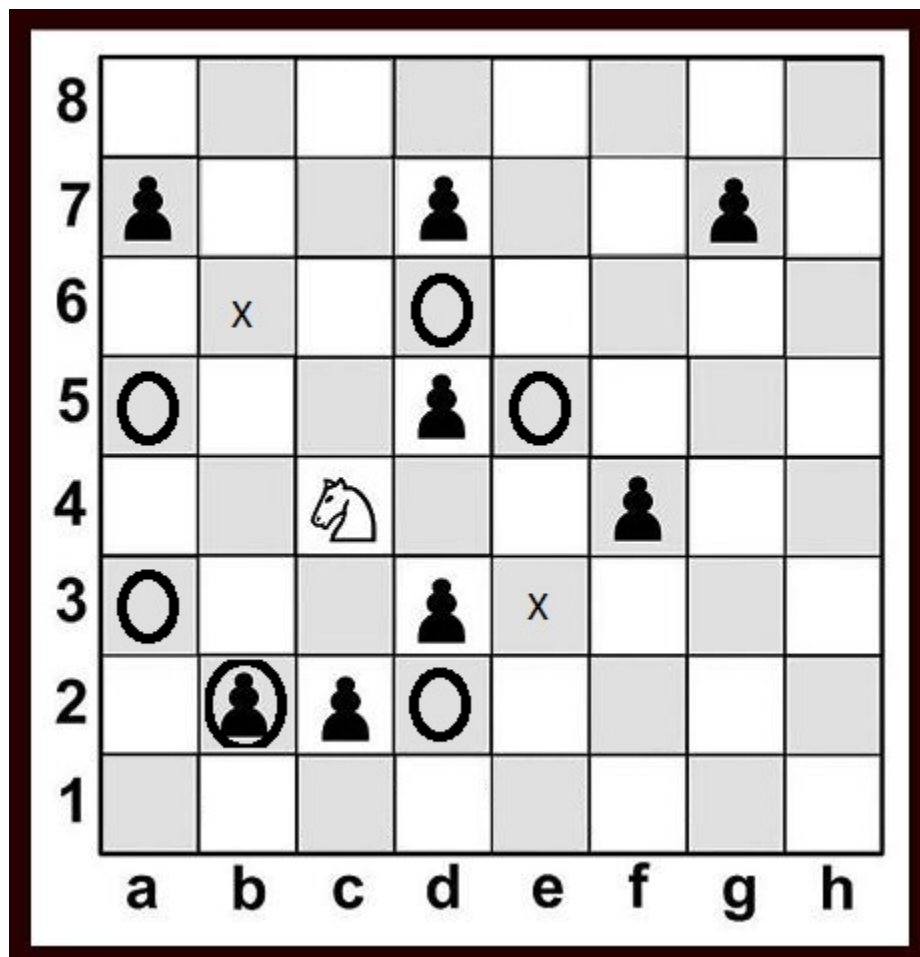
URI Online Judge | 1147

Fuga do Cavalo

Por Gerson Groth  Brasil

Timelimit: 1

Seu amigo Pedro está aprendendo a jogar xadrez. Mas ele ainda não tem segurança de como pode movimentar o Cavalo. Desta forma, ele solicitou a você que desenvolvesse um programa que calcule, em apenas uma jogada, o número de distintos movimentos que o cavalo pode realizar, sem que o mesmo fique sobre o ataque de qualquer um dos 8 peões. Os movimentos do cavalo e dos peões são realizados conforme o jogo de xadrez tradicional, ou seja, o cavalo pode somente andar em “L”, e os peões atacar somente para frente em diagonal. Segue abaixo um exemplo:



Na imagem apresentada, das 8 distintas posições que o cavalo pode movimentar-se no tabuleiro, 2 delas estão sobre ataque (6b e 3e). Nas demais casas, o cavalo pode movimentar-se seguramente, conseguindo fugir do ataque de qualquer um dos peões. Note que na casa 2b já existe um peão, porém ainda é uma jogada válida, já que o cavalo pode movimentar-se nessa mesma casa e “matar” o peão.

Portanto, para o exemplo apresentado, o número de movimentos seguros para o cavalo realizar são 6. Vale

lembrar que os peões pretos somente podem se movimentar de cima para baixo no tabuleiro, ou seja, da linha 8 até a linha 1, não sendo permitido o seu retorno.

Entrada

A entrada consiste de diversos casos de teste. Cada caso de teste consiste em 9 linhas de entrada. A primeira linha indica a posição inicial do cavalo. As demais 8 linhas, representam as respectivas posições dos peões.

O final da entrada consiste em uma única linha contendo somente o valor 0 (zero).

Saída

Para cada teste de caso de entrada, seu programa deve imprimir uma única linha, contendo a seguinte descrição:


Caso de Teste #Y: X movimento(s).

onde Y representa o número do respectivo caso de teste, e X representa a quantidade de movimentos possíveis ao cavalo realizar, em uma única rodada, sem que fique sobre ataque de um ou mais peões.

Exemplo de Entrada	Exemplo de Saída
4c 2b 2c 3d 4f 5d 7a 7d 7g 0	Caso de Teste #1: 6 movimento(s).

URI Online Judge | 1091

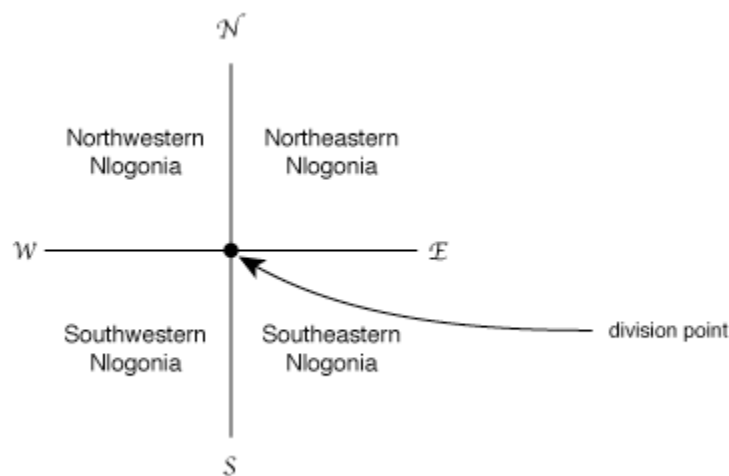
Divisão da Nlogônia

Por Ricardo Anido  Brasil

Timelimit: 1

Depois de séculos de escaramuças entre os quatro povos habitantes da Nlogônia, e de dezenas de anos de negociações envolvendo diplomatas, políticos e as forças armadas de todas as partes interessadas, com a intermediação da ONU, OTAN, G7 e SBC, foi finalmente decidida e aceita por todos a maneira de dividir o país em quatro territórios independentes.

Ficou decidido que um ponto, denominado ponto divisor, cujas coordenadas foram estabelecidas nas negociações, definiria a divisão do país, da seguinte maneira. Duas linhas, ambas contendo o ponto divisor, uma na direção norte-sul e uma na direção leste-oeste, seriam traçadas no mapa, dividindo o país em quatro novos países. Iniciando no quadrante mais ao norte e mais ao oeste, em sentido horário, os novos países seriam chamados de Nlogônia do Noroeste, Nlogônia do Nordeste, Nlogônia do Sudeste e Nlogônia do Sudoeste.



A ONU determinou que fosse disponibilizada uma página na Internet para que os habitantes pudessem consultar em qual dos novos países suas residências estão, e você foi contratado para ajudar a implementar o sistema.

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um inteiro K indicando o número de consultas que serão realizadas ($0 < K \leq 10^3$). A segunda linha de um caso de teste contém dois números inteiros N e M representando as coordenadas do ponto divisor ($-10^4 < N, M < 10^4$). Cada uma das K linhas seguintes contém dois inteiros X e Y representando as coordenadas de uma residência ($-10^4 \leq X, Y \leq 10^4$). Em todas as coordenadas dadas, o primeiro valor corresponde à direção leste-oeste, e o segundo valor corresponde à direção norte-sul.

O final da entrada é indicado por uma linha que contém apenas o número zero.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma linha contendo:

- a palavra *divisa* se a residência encontra-se em cima de uma das linhas divisórias (norte-sul ou leste-oeste);
- NO se a residência encontra-se na Nlogônia do Noroeste;
- NE se a residência encontra-se na Nlogônia do Nordeste;
- SE se a residência encontra-se na Nlogônia do Sudeste;
- SO se a residência encontra-se na Nlogônia do Sudoeste.

Exemplo de Entrada	Exemplo de Saída
3	NE
2 1	divisa
10 10	NO
-10 1	divisa
0 33	NE
4	SO
-1000 -1000	SE
-1000 -1000	
0 0	
-2000 -10000	
-999 -1001	
0	

Maratona de Programação da SBC 2008.

Alarme Despertador

Maratona de Programação da SBC 2009  Brasil

Timelimit: 1

Daniela é enfermeira em um grande hospital, e tem os horários de trabalho muito variáveis. Para piorar, ela tem sono pesado, e uma grande dificuldade para acordar com relógios despertadores.

Recentemente ela ganhou de presente um relógio digital, com alarme com vários tons, e tem esperança que isso resolva o seu problema. No entanto, ela anda muito cansada e quer aproveitar cada momento de descanso. Por isso, carrega seu relógio digital despertador para todos os lugares, e sempre que tem um tempo de descanso procura dormir, programando o alarme despertador para a hora em que tem que acordar. No entanto, com tanta ansiedade para dormir, acaba tendo dificuldades para adormecer e aproveitar o descanso.

Um problema que a tem atormentado na hora de dormir é saber quantos minutos ela teria de sono se adormecesse imediatamente e acordasse somente quando o despertador tocasse. Mas ela realmente não é muito boa com números, e pediu sua ajuda para escrever um programa que, dada a hora corrente e a hora do alarme, determine o número de minutos que ela poderia dormir.

Entrada

A entrada contém vários casos de teste. Cada caso de teste é descrito em uma linha, contendo quatro números inteiros H_1 , M_1 , H_2 e M_2 , com $H_1:M_1$ representando a hora e minuto atuais, e $H_2:M_2$ representando a hora e minuto para os quais o alarme despertador foi programado ($0 \leq H_1 \leq 23$, $0 \leq M_1 \leq 59$, $0 \leq H_2 \leq 23$, $0 \leq M_2 \leq 59$).

O final da entrada é indicado por uma linha que contém apenas quatro zeros, separados por espaços em branco.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma linha, cada uma contendo um número inteiro, indicando o número de minutos que Daniela tem para dormir.

Sample Input	Sample Output
1 5 3 5	120
23 59 0 34	35
21 33 21 10	1417
0 0 0 0	

URI Online Judge | 1271

Onde estão meus Genes

Por Vinícius Fortuna  Brasil

Timelimit: 3

Uma maneira que os cientistas tem para tentar medir como uma espécie evoluiu para outra é investigando como o genoma do ancestral se modificou para se transformar nesta outra espécie. Espécies intimamente relacionadas têm vários genes em comum e verifica-se que uma boa maneira de compará-las é através da comparação de como os genes comuns mudaram de lugar.

Uma das mutações mais comuns que alteram a ordem dos genes de genomas é a inversão. Se modelarmos um genoma como uma sequência de N genes sendo cada gene um número inteiro de 1 a N , então uma inversão é uma mutação que altera o genoma revertendo a ordem de um bloco de genes consecutivos. A inversão pode ser descrita por dois índices (i, j) , $(1 \leq i \leq j \leq N)$, indicando que ela inverte a ordem dos genes dentro de índices de i até j .

Assim, quando isto é aplicado para um genoma $[g_1, \dots, g_{i-1}, g_i, g_{i+1}, \dots, g_{j-1}, g_j, g_{j+1}, \dots, g_N]$, obtém-se o genoma $[g_1, \dots, g_{i-1}, g_j, g_{j-1}, \dots, g_{i+1}, g_i, g_{j+1}, \dots, g_N]$. Como um exemplo, a inversão de $(3, 6)$, aplicado à genoma $[1, 2, 3, 4, 5, 6, 7]$ dá $[1, 2, 6, 5, 4, 3, 7]$. Se depois que a inversão $(1, 3)$ é aplicada, obtém-se o genoma $[6, 2, 1, 5, 4, 3, 7]$.

Um cientista que está estudando a evolução de uma espécie deseja tentar uma série de inversões no genoma desta espécie. Em seguida, ele quer consultar a posição final de vários genes. Será que você aceita o desafio de ajudá-lo?

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um inteiro N indicando o número de genes no genoma ($1 \leq N \leq 50000$). Você pode supor que o ordem inicial dos genes é a sequência de números inteiros de 1 a N em ordem crescente. A segunda linha de um caso de teste contém um inteiro R ($0 \leq R \leq 1000$) que indica o número de inversões a serem aplicadas ao genoma. Então, R linhas seguem, cada uma contendo dois inteiros i, j ($1 \leq i \leq j \leq N$), separados por um único espaço, o qual indicam os dois índices que definem a inversão correspondente. Após a descrição das inversões há uma linha contendo um inteiro Q ($0 \leq Q \leq 100$), que indica o número de consultas para os genes, seguido de Q linhas, onde cada linha contém um inteiro representando um gene cuja posição final você deve determinar.

O final da entrada é indicada por $N = 0$.

Saída

Para cada caso de teste da entrada seu programa deve produzir $Q + 1$ linhas de saída. A primeira linha deve conter a string "**Genome**", seguido do número do caso de teste. As seguintes Q linhas devem conter um número inteiro, cada um representando as respostas das consultas.

Exemplo de Entrada

Exemplo de Saída

Exemplo de Entrada	Exemplo de Saída
9	Genome 1
1	1
3 6	6
4	4
1	1
3	Genome 2
5	1
1	5
5	
2	
1 2	
1 5	
2	
5	
2	
0	

ACM/ICPC South America Contest 2005