



Universidade Federal de Pernambuco  
Centro de Informática



# Final da Seletiva 2016

## Maratona de Programação

9 de Abril de 2016

### Caderno de Problemas

Este caderno contém 10 problemas; as páginas estão numeradas de 1 a 15, contando esta página de rosto. Verifique se o caderno está completo.

### Informações Gerais

#### A) Sobre a entrada

- 1) A entrada do seu programa deve ser lida da *entrada padrão (stdin)*.
- 2) A entrada é composta por vários casos de teste, cada um descrito em um número de linhas que depende do problema. A primeira linha da entrada é sempre o número de casos de teste.
- 3) Quando uma linha da entrada contém vários valores, estes serão separados por um único espaço em branco. Não haverá nenhum outro espaço em branco.
- 4) Cada linha, incluindo a última, contém o caractere final-de-linha.

#### B) Sobre a saída

- 1) A saída do seu programa deve ser escrita na *saída padrão (stdout)*.
- 2) Quando uma linha da saída contém vários valores, estes devem ser separados por um único espaço em branco. Não deverá haver nenhum outro espaço em branco.
- 3) Cada linha, incluindo a última, deve conter o caractere final-de-linha.

## Problema A

# Álibi

Um crime grave ocorreu, e  $N$  pessoas prestaram depoimento na *International Criminal Pernambuco Court* (ICPC). O  $i$ -ésimo indivíduo, em seu depoimento, pode ter citado o  $j$ -ésimo indivíduo, se tornando álibi do mesmo, ou pode não ter citado ninguém. Em outras palavras, no caso do  $i$ -ésimo indivíduo citar, em seu depoimento, a  $j$ -ésima pessoa, o indivíduo  $i$  se torna álibi do indivíduo  $j$ . Dado a lista das pessoas citadas em cada depoimento, calcule a quantidade de pessoas que não possuem um álibi, e a quantidade de pessoas *suspeitas*. Uma pessoa  $i$  é *suspeita* se, e somente se,  $i$  não tem álibi, o único álibi de  $i$  é o próprio  $i$ , ou **todos** os álibis  $j$  de  $i$  que sejam diferentes do próprio  $i$  são *suspeitos*.

### Entrada

A primeira linha da entrada contém  $T$  ( $1 \leq T \leq 200$ ), o número de casos de teste. Cada caso começa com um inteiro  $N$  ( $1 \leq N \leq 7555$ ), indicando a quantidade de indivíduos que prestaram depoimento. A seguir, haverá uma linha com  $N$  inteiros. Cada inteiro  $x_i$  ( $0 \leq x_i \leq N$ ) indica que a  $i$ -ésima pessoa citou o  $x$ -ésimo indivíduo, ou 0 se a  $i$ -ésima pessoa não citou ninguém.

### Saída

Para cada caso de teste imprima “Caso # $T$ :  $X$   $Y$ ”, onde  $T$  é o número do caso de teste, começando em 1,  $X$  é a quantidade de indivíduos sem álibi, e  $Y$  é a quantidade de indivíduos *suspeitos*, segundo a definição da questão.

Exemplo de Entrada	Exemplo de Saída
2	Caso #1: 1 2
2	Caso #2: 2 3
2 0	
6	
1 3 4 2 2 0	

No primeiro caso, o indivíduo 1 afirma que é álibi do indivíduo 2, indivíduo 2 não citou ninguém em seu depoimento. O indivíduo sem álibi é o indivíduo 1, e todos são suspeitos.

No segundo caso, os indivíduos sem álibi são o 5 e o 6, e os indivíduos suspeitos são 1, 5 e 6.

Autor: *Israel Batista*

## Problema B

# Bar do Pinto

O Bar do Pinto é um local onde ocorrem muitas festividades, e assim como muitas disciplinas, ele também possui um monitor para auxiliar os alunos. O deste período é o astuto Renatinho, estudante de engenharia do Centro de Informática. Durante as comemorações de recepção aos novos ingressantes da UFPE, ele passa um desafio para os alunos, e o aluno que acertar ganhará uma bolsa de iniciação científica no bar, podendo realizar pesquisas nas áreas de Computação Efílica, Passação Computacional e Interação com Crush.



Renatinho formulou o seguinte desafio:

“Neste bar existem  $N$  drinks, enumerados de 1 até  $N$ . O  $i$ -ésimo drink custa  $P_i$  reais, e não se pode comprar o mesmo drink duas vezes. São colocadas  $M$  moedas enfileiradas, a primeira moeda está no índice 1 e a última no índice  $M$  desta lista, a  $i$ -ésima moeda vale  $Q_i$  reais. Você pode escolher qualquer sequência contígua de moedas, passando a possuir a soma dos valores das moedas nesse intervalo e usá-las para comprar os drinks, seu objetivo é que com as moedas escolhidas, seja possível comprar no mínimo  $K$  drinks.

Qual o tamanho da menor subsequência que satisfaz os critérios definidos?

O tamanho de uma subsequência contígua que contém as moedas no intervalo  $[L, R]$  é definida com  $R-L+1$ .”

Elabore um programa que, dado o número de drinks  $N$ , o custo de cada drink, o número de moedas  $M$ , o valor de cada moeda, e um inteiro  $K$ , responda o tamanho mínimo da sequência contígua que satisfaça os critérios do desafio para ganhar a premiação oferecida por Renatinho.

### Entrada

A primeira linha da entrada contém  $T$  ( $1 \leq T \leq 100$ ), o número de casos de teste. Cada caso de teste começa com uma linha contendo três inteiros  $N$ ,  $M$  e  $K$  ( $1 \leq N \leq 10^6$ ,  $1 \leq M \leq 10^6$ ,  $1 \leq K \leq N$ ), definindo a quantidade de drinks e moedas, respectivamente. A próxima linha contém  $N$  inteiros  $P_i$  indicando o custo do  $i$ -ésimo drink ( $1 \leq i \leq N$ ,  $1 \leq P_i \leq 10^3$ ), seguindo de uma linha contendo  $M$  inteiros  $Q_i$  indicando o valor da  $i$ -ésima moeda ( $1 \leq i \leq M$ ,  $1 \leq Q_i \leq 10^3$ ).

### Saída

Para cada caso de teste imprima “Caso # $T$ :  $X$ ”, onde  $T$  é o número do caso de teste, começando em 1, e  $X$  é o tamanho mínimo da sequência. Caso não exista sequência que satisfaça os critérios, então imprima -1 no lugar de  $X$ .

Exemplo de Entrada	Exemplo de Saída
3	Caso #1: 1
3 2 2	Caso #2: 3

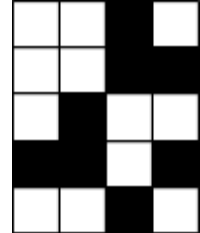
1 5 10 10 5 2 5 2 7 10 10 1 7 9 7 2 5 1 30 30 1 2 3 4 5	Caso #3: -1
--	-------------

Autor: *Marlon Reghert*

## Problema C

# Caixas

Teobaldo recebeu de presente de aniversário um grid  $N \times M$  e um bolo. Algumas células do grid são ocupadas por caixas, enquanto outras estão vazias. Porém, ele só poderia comer o bolo caso ele respondesse corretamente a seguinte pergunta: quantos retângulos distintos existem no grid tal que todas as células do retângulo são ocupadas por caixas? Um retângulo de dimensões  $A \times B$  é diferente de um retângulo de dimensões  $C \times D$  caso  $A \neq C$  ou  $B \neq D$ . Os retângulos não podem ser rotacionados. Você, como nobre convidado do aniversário de Teobaldo, foi chamado para ajudá-lo a responder corretamente a pergunta. Como presente, você receberá a maior fatia do bolo!



No exemplo ao lado, há 5 retângulos distintos que são completamente formados por caixas, como mostrado a seguir:



### Entrada

A primeira linha da entrada contém  $T$  ( $1 \leq T \leq 100$ ), o número de casos de teste. Cada caso de teste começa com uma linha contendo dois inteiros  $N$  e  $M$  ( $1 \leq N, M \leq 1000$ ), as dimensões do grid de caixas. As  $N$  linhas seguintes contém  $M$  caracteres cada uma, sem espaço entre si, sendo 'C' para indicar a presença de caixa naquela posição, e 'X' para indicar a ausência.

### Saída

Para cada caso de teste imprima “Caso # $T$ :  $X$ ”, onde  $T$  é o número do caso de teste, começando em 1,  $X$  é o número de retângulos de caixas de dimensões distintas que Teobaldo consegue formar com o grid dado.

Exemplo de Entrada	Exemplo de Saída
2	Caso #1: 0
1 1	Caso #2: 5
X	
5 4	
CCXC	
CCXX	
CXCC	
XXCX	
CCXC	

Autor: *Gustavo Stor*

## Problema D

# Dados

Ana e Bob estão sem nada pra fazer e resolveram pegar dois dados para brincar. A ideia do jogo é bem simples, cada um começa com 0 moedas e o jogo consiste de várias rodadas, em cada rodada, cada um dos jogadores pega um dado e joga. Quem fizer mais pontos ganha todas as moedas do oponente, ou seja, o oponente fica com 0 moedas. Caso seja empate, ambos ganham 1 moeda do amigo deles Tito (Tito é muito legal e possui um milhão de moedas). Para eles não precisarem ficar contando com quantas moedas cada um está em um momento, seria bem legal um programa que fizesse isso. Eles vão jogar os dados **N** vezes, na **i**-ésima vez Ana consegue o valor **A** e Bob consegue o valor **B**. Com essas informações, você consegue fazer um programa que diga quantas moedas Ana e Bob possuem ao final das **N** rodadas?

### Entrada

A primeira linha da entrada contém **T** ( $1 \leq T \leq 50$ ), o número de casos de teste. A primeira linha de cada caso de teste contém **N** ( $1 \leq N \leq 10^4$ ), o número de rodadas. Cada uma das **N** linhas seguintes contém dois inteiros **A** ( $1 \leq A \leq 6$ ) e **B** ( $1 \leq B \leq 6$ ), representando o valor obtido por Ana e por Bob, respectivamente, naquela rodada.

### Saída

Para cada caso de teste imprima “Caso #**T**: **X Y**”, onde **T** é o número do caso de teste, começando em 1, **X** é a quantidade de moedas de Ana e **Y** é a quantidade de moedas de Bob, após as **N** rodadas.

Exemplo de Entrada	Exemplo de Saída
3	Caso #1: 0 0
2	Caso #2: 4 0
1 5	Caso #3: 2 2
6 2	
3	
1 1	
6 6	
4 3	
2	
4 4	
5 5	

Autor: *Duhan Caraciolo*

## Problema E

# Ednaldo Pereira

O nome dele é Pereira. Ednaldo Pereira. Ele gosta muito de strings, principalmente aquelas que se parecem com “Ednaldo Pereira”. Ednaldo estuda informática teórica, e foca em conceitos sobre linguagens e relações entre cadeias. Pereira definiu que duas cadeias  $A$  e  $B$  são *Ednaldo-equivalentes* se:

- elas têm o mesmo comprimento  $N$ ;
- para todo  $0 \leq i < N$ ,  $A[i] = B[i]$  ou  $A[i]$  e  $B[i]$  representam a mesma letra, só que uma é maiúscula e a outra minúscula, ou vice-versa.

Ednaldo Pereira, também conhecido como Ednaldo Pereira, também gosta de funções de hash, e inventou a função *Pereira-hash*: dada uma string  $S$  e um array de números reais **VAL** de tamanho 256, o valor do *Pereira-hash* de  $S$  é igual ao somatório, para cada caractere  $c_i$  de  $S$ , de  $\text{VAL}[c_i]$ , onde o valor de  $c_i$  em ASCII é usado como índice no array **VAL**.

Ednaldo “Ednaldo Pereira” Pereira escolheu cuidadosamente os valores de **VAL** para usar a função *Pereira-hash* em seu novo algoritmo de encriptação, o Algoritmo Ednaldo-Pereira. Com ele, Ednaldo encriptou suas novas músicas (sim, ele também é músico), na esperança de que ninguém tivesse a chance de roubá-las. Mas sua ex-mulher, contrariada com o sucesso de Pereira nos estudos e na música, jogou fora todo o array **VAL**. Agora Ednaldo precisa de sua ajuda para desencriptar suas novas músicas, sem as quais sua carreira não vale nada. Vale tudo para ajudá-lo!

O que Ednaldo, O Pereira, quer é o seguinte: dada uma string  $S$ , Ednaldo quer saber quantas strings *Ednaldo-equivalentes* a  $S$  existem que também possuam o mesmo valor de *Pereira-hash* de  $S$ , independentemente dos valores do array **VAL**. Pereira confia nas suas habilidades de programador para achar a quantidade total de strings que têm essas propriedades para quaisquer valores de **VAL**, dada a string  $S$  e as definições que Ednaldo lhe ensinou. Não trate Pereira com indiferença, ajude-o, e mostre a Ednaldo qual é o irmão!

### Entrada

A primeira linha da entrada contém  $T$  ( $1 \leq T \leq 100$ ), o número de casos de teste.

Em seguida, a única linha de cada caso de teste contém uma string  $S$  de tamanho  $N$  ( $1 \leq N \leq 100$ ).  $S$  pode conter apenas letras maiúsculas ou minúsculas e o caractere underline (`'_'`).

### Saída

Para cada caso de teste imprima em uma única linha “Caso # $T$ :  $X$ ”, onde  $T$  é o número do caso atual, começando em 1, e  $X$  é a quantidade de strings existentes que são *Ednaldo-equivalentes* a  $S$  e que também têm o mesmo valor de *Pereira-hash*. A string de entrada não deve ser considerada como parte da resposta. É garantido que o valor de  $X$  para cada caso não excede  $10^9$ .

Exemplo de Entrada	Exemplo de Saída
4	Caso #1: 3
EdNaLDo_pErElrA	Caso #2: 0
Mesclado	Caso #3: 11
GoD_is_gOOd	Caso #4: 8
_homem_OFERECIDO_	

No primeiro caso, as strings existentes são:

EdNALDo\_pErElra

EDNaLdo\_pErElrA

EDNALdo\_pErElra

Não existe string *Ednaldo-equivalente* e com o mesmo *Pereira-hash* no segundo caso.

Autor: *Ednaldo Pereira*



## Problema F

# Ferrou!

Tom está prestes a viajar de avião. Como a maioria das pessoas sabem, existe um período do voo, quando o avião está abaixo de  $X$  metros, em que todos os eletrônicos devem ser desligados, não podendo ser usado nem mesmo em modo avião. Porém, Tom é mal-educado e não gosta de respeitar regras. A companhia aérea, sabendo que Tom está prestes a embarcar, quer registrar quantas vezes Tom vai desrespeitar a regra para que possa puni-lo de alguma forma. Para que a companhia possa tomar medidas um pouco mais complexas, ela irá registrar, para cada um dos  $N$  minutos totais do voo, a altura do voo atual e se Tom está ou não usando o celular. A companhia não quer apenas saber quantas vezes Tom descumpriu a regra durante todo o voo, mas também quantas vezes ele descumpriu a regra no período do minuto  $A$  até o minuto  $B$  do voo, ou seja, quantas vezes Tom usou o celular enquanto o avião estava abaixo de  $X$  metros, entre os minutos  $A$  e  $B$ , inclusive. A companhia irá verificar  $M$  desses períodos. Será que Tom se ferrou?

### Entrada

A primeira linha da entrada contém  $T$  ( $1 \leq T \leq 10$ ), o número de casos de teste. A primeira linha de cada caso de teste contém três inteiros  $N$  ( $1 \leq N \leq 10^6$ ),  $X$  ( $1 \leq X \leq 10^9$ ) e  $M$  ( $1 \leq M \leq 10^4$ ). A segunda linha de cada caso de teste contém  $N$  números separados por um espaço, onde o  $i$ -ésimo número ( $1 \leq \text{número} \leq 10^9$ ) representa a altura em que o avião estava no  $i$ -ésimo minuto do voo.

A terceira linha de cada caso de teste contém uma string com  $N$  caracteres, que podem ser 0 ou 1. Se o  $i$ -ésimo caractere for 0, então Tom não estava com o celular ligado no  $i$ -ésimo minuto e, caso seja 1, ele estava. As próximas  $M$ , e últimas, linhas de cada caso de teste contém dois inteiros  $A$  e  $B$  ( $1 \leq A \leq B \leq N$ ), o intervalo do voo que a companhia está interessada.

### Saída

Para cada caso de teste imprima em uma única linha "Caso # $T$ :", onde  $T$  é o número do caso atual, começando em 1. Em seguida, para cada um dos  $M$  intervalos do caso, na ordem dada na entrada, imprima quantas vezes Tom desrespeitou a regra naquele intervalo.

Exemplo de Entrada	Exemplo de Saída
2	Caso #1:
5 10 2	2
23 5 2 10 11	2
11111	Caso #2:
1 5	1
2 4	0
4 200 2	
100 200 300 400	
1101	
1 3	
3 4	

## Problema G

# Gato Teobaldo

Teobaldo é um gato que se acha muito humano. Por se achar um humano, ele adora comer, dormir, estudar e brincar, principalmente com Guga. Tem uma brincadeira em particular que ele adora: Guga diz para Teobaldo uma palavra **S** e, em seguida faz **M** perguntas do tipo: qual a quantidade de subsequências **distintas** de **S** que contém prefixo **X**. Teobaldo só reconhece os caracteres que pertencem as letras do alfabeto brasileiro e que são minúsculas, logo Guga não irá trapacear e oferecer uma palavra que Teobaldo não reconheça. Guga, apesar de ser um exímio programador, pede sua ajuda para fazer este cálculo e poder continuar brincando com Teobaldo e desenvolvendo o Maratonando.



Para deixar claro, Guga diz que uma palavra **S** é uma subsequência de uma palavra **P** se a única operação aplicável em **P** para chegar em **S** é remover caracteres.

### Entrada

A primeira linha da entrada contém um inteiro **T** ( $1 \leq T \leq 20$ ) que corresponde ao número de casos de teste. A primeira linha de cada caso de teste contém a palavra **S** ( $1 \leq |S| \leq 20$ ) e um inteiro **M** ( $1 \leq M \leq 10^4$ ), sendo **S** a *string* dada por Guga e **M** a quantidade de perguntas feitas a Teobaldo, separados por um único espaço. As **M** seguintes linhas contém uma única palavra **X** ( $1 \leq |X| \leq |S|$ ) por linha, sendo **X** o prefixo de cada pergunta.

### Saída

Para cada caso de teste imprima em uma única linha "Caso #**T**:", onde **T** é o número do caso atual, começando em 1. Então imprima **M** linhas, sendo **M** o número de perguntas daquele caso, e cada linha *i* contendo um inteiro correspondente a resposta de Teobaldo para a *i*-ésima pergunta.

Exemplo de Entrada	Exemplo de Saída
2	Caso #1:
banana 3	4
ana	7
ban	0
salsa	Caso #2:
bbaaaa 3	4
a	3
aa	4
ba	

Autor: Victor Monteiro

## Problema H

# Horário dos Ônibus

Uma pesquisa do Instituto dos Calmos, Pacíficos e Comedidos (ICPC) descobriu recentemente que os alunos universitários que usam o transporte público tem uma maior tendência de ficar irritados. O cálculo do risco de ficar irritado envolve, entre outras coisas, o tempo de deslocamento de sua casa até a universidade.

Na cidade deste problema, existem terminais de ônibus que possuem linhas de ônibus que levam as pessoas de um terminal a outro. Cada linha de ônibus  $A_i/B_i$  sai do terminal  $A_i$  em intervalos fixos de  $X_i$  minutos e leva os passageiros até o terminal  $B_i$  em  $Y$  minutos — para fins de simplicidade,  $Y$  é fixo para todas as linhas de ônibus da cidade. Um aluno universitário precisa percorrer um subconjunto de terminais de ônibus, a partir de um terminal de origem no minuto zero, para chegar ao seu destino. Considere que no tempo zero todas as linhas de ônibus acabaram de partir dos terminais e o aluno embarcou em alguma linha no terminal de origem. Como você é um programador destemido, sua ajuda foi pedida para encontrar o menor coeficiente de raiva (o tempo para chegar no terminal mais próximo da universidade) possível que um aluno pode adquirir em seu caminho entre dois terminais.

### Entrada

A primeira linha da entrada possui um inteiro  $T$  ( $1 \leq T \leq 70$ ), representando a quantidade de casos de teste. A primeira linha de cada caso de teste possui duas strings  $O$  e  $D$  ( $O \neq D$ ), o nome do terminal de origem e o nome do terminal de destino, respectivamente, seguidas por dois inteiros,  $Y$  ( $1 \leq Y \leq 100$ ), indicando o tempo da viagem entre dois terminais, e  $M$  ( $0 \leq M \leq 10^5$ ), o número de linhas de ônibus. Cada linha  $i$  das próximas  $M$  linhas contém um nome  $A_i$ , um nome  $B_i$  e um inteiro  $X_i$  ( $A_i \neq B_i$ ,  $1 \leq X_i \leq 100$ ), indicando que existe uma linha de ônibus que sai a cada  $X_i$  minutos do terminal  $A_i$  até o terminal  $B_i$ . É garantido que não existe mais do que 1000 terminais por caso de teste e cada um de seus nomes é definido por uma palavra de 1 a 20 letras maiúsculas ou minúsculas.

### Saída

Para cada caso de teste imprima em uma única linha "Caso # $T$ :  $Z$ ", onde  $T$  é o número do caso atual, começando em 1, e  $Z$  é o menor coeficiente de raiva possível para o problema proposto, seguido de um espaço, seguido da palavra "anticalmas", que é a unidade deste coeficiente. Ou, se não for possível alcançar o destino, imprima "Destino inalcançavel" no lugar de  $Z$ .

Exemplo de Entrada	Exemplo de Saída
2 Cavaleiro Tancredo 50 5 Cavaleiro Barro 12 Barro Macaxeira 30 Macaxeira Tancredo 15 Barro Joana 5 Joana Tancredo 24 Cavaleiro Tancredo 50 5 Cavaleiro Barro 12	Caso #1: 170 anticalmas Caso #2: 162 anticalmas

Barro Macaxeira 30 Macaxeira Tancredo 14 Barro Joana 5 Joana Tancredo 24	
---	--

Explicação do **segundo** exemplo:

Rota 1: 50 minutos de Cavaleiro para Barro + 10 minutos de espera pelo Barro/Macaxeira + 50 minutos de Barro para Macaxeira + 2 minutos de espera pelo Macaxeira/Tancredo + 50 minutos de Macaxeira para Tancredo. Coeficiente de raiva = 162 anticalmas

Rota 2: 50 minutos de Cavaleiro para Barro + 0 minutos de espera pelo Barro/Joana + 50 minutos de Barro para Joana + 20 minutos de espera pelo Joana/Tancredo + 50 minutos de Joana para Tancredo. Coeficiente de Raiva = 170 anticalmas

Autor: Rogério Santos

## Problema I

# Itens de Caruaru

Dudu está voltando dos Estados Unidos e pretende comprar, no máximo, **M** lembrancinhas na feira de Caruaru para matar as saudades do seu querido estado. Na feira, há **N** itens que podem ser comprados. Dudu organizou estes itens em **G** grupos, de maneira que todos os itens em um grupo são similares e, assim sendo, Dudu comprará, no máximo, um item de cada grupo. Dudu tem **X** reais para gastar com estas compras e quer gastar o máximo, obedecendo todas as restrições anteriores. Ajude Dudu a calcular o quanto ele gastará com estas compras.

### Entrada

A primeira linha da entrada contém **T** ( $1 \leq T \leq 100$ ), o número de casos de teste. Cada caso de teste começará com 4 inteiros, **N**, **M**, **G** e **X** ( $1 \leq M, G \leq N \leq 200$ ,  $0 \leq X \leq 300$ ). As próximas **G** linhas listarão quantos itens estão no grupo, e qual o valor de cada um dos itens. A soma das quantidades de todos os grupos será igual à **N**.

### Saída

Para cada caso de teste imprima em uma única linha “Caso #**T**: **X**”, onde **T** é o número do caso atual, começando em 1, e **X** é o quanto Dudu irá gastar com suas compras.

Exemplo de Entrada	Exemplo de Saída
3	Caso #1: 72
5 2 3 100	Caso #2: 83
2 20 22	Caso #3: 0
2 10 11	
1 50	
5 3 3 100	
2 20 22	
2 10 11	
1 50	
2 1 1 100	
2 101 110	

No primeiro caso, há 5 itens divididos em 3 grupos e Dudu escolherá 2 itens para comprar. Os itens do primeiro grupo custam 20 e 22 reais, os itens do segundo grupo custam 10 e 11 reais, e o item do terceiro grupo custa 50 reais. Na solução ótima, Dudu escolhe os itens que custam 50 e 22 reais.

Autor: *Israel Batista*

## Problema J

# Jarathus, o Conquistador

Jarathus é um grande conquistador, sendo reconhecido por ter alcançado a marca do maior império de Askard jamais visto. Segundo a lenda, Jarathus conquistou 40 reinos do Grande Continente no período de um mês. Agora, restam apenas poucos reinos para uma total hegemonia, o que será o sinal definitivo da sua capacidade de liderança e de conquistas. Será Jarathus o rei dos conquistadores?

Mas bem, sabe-se que toda vez que Jarathus conquista um reino, ele gasta a quantia necessária para a invasão, mas depois arrecada impostos dos reinos conquistados anteriormente, segundo o contrato de submissão. Quanto maior a inimizade entre o reino recém conquistado e os anteriores, maior a arrecadação de impostos.

Por exemplo, dada a seguinte tabela de inimizades:

	A	B	C
A'	0	5	10
B'	2	0	0
C'	5	2	0

Se Jarathus conquistar o reino A possuindo os reinos B e C, arrecadará 15. Se conquistar o reino B, possuindo o Reino A, arrecadará 2, e assim sucessivamente.

### Entrada

A primeira linha da entrada contém um inteiro  $T$  ( $1 \leq T \leq 100$ ) que corresponde ao número de casos de teste. Em cada caso de teste, a primeira linha contém um inteiro  $N$  ( $1 \leq N \leq 15$ ), indicando o número de reinos que Jarathus precisa conquistar. A segunda linha contém  $N$  inteiros, indicando o custo  $C_i$  ( $-100 \leq C_i \leq 100$ ) de invadir o  $i$ -ésimo reino. As seguintes  $N$  linhas contém  $N$  inteiros  $a_{ij}$  ( $-100 \leq a_{ij} \leq 100$ ) cada, indicando a inimizade do reino  $i$  para com o reino  $j$  (não necessariamente  $a_{ij} = a_{ji}$ ).

### Saída

Para cada caso de teste imprima em uma única linha "Caso # $T$ :", onde  $T$  é o número do caso atual, começando em 1, e  $X$  é o montante máximo após conquistar todos os reinos.

Exemplo de Entrada	Exemplo de Saída
2	Caso #1: 13
3	Caso #2: 10
0 1 3	
0 5 10	
2 0 0	
5 2 0	
5	
0 0 0 0 0	
0 1 1 1 1	
0 0 1 1 1	

0 0 0 1 1 0 0 0 0 1 0 0 0 0 0	
-------------------------------------	--

Autor: *Giovanni Barros*