

---

Seletiva para  
Maratona de Programação de 2008

Universidade de São Paulo

Caderno de Problemas

Departamento de Ciência da Computação IME-USP  
Domingo, 17 de agosto de 2008.

---

## Problema A: Seqüências de Röntgen

Arquivo: *rontgen*. [c/cpp/java]

Wilhelm Conrad Röntgen foi um físico alemão que viveu no final do século XIX e início do século XX. Suas experiências em radiação eletromagnéticas renderam a ele o primeiro prêmio Nobel em Física, outorgado em 1901. Sua principal descoberta foi a existência do “raio X” e seu uso em aplicações médicas. Em 22 de dezembro de 1895, Röntgen fez um raio X da mão de sua esposa (com um anel em um dos dedos). A descoberta do raio X causou grande alvoroço na época e já em 1896 jornais europeus noticiavam a invenção e as grandes possibilidades de enxergar por dentro dos corpos sem a necessidade de cortá-los. A morte de Röntgen, causada por um certo tipo de câncer, é atribuída às radiações constantes a que esteve exposto durante suas pesquisas científicas.

Röntgen começou a desconfiar da existência de radiações invisíveis quando, nas suas pesquisas, era capaz de medir alterações consideráveis na fluorescência dos objetos quando colocados num tubo de Lenard que era submetido a uma corrente elétrica. Os estudos de Röntgen foram tão precisos que ele pôde inclusive gerar a seqüência que era observada no tubo de Lenard em cada instante de tempo. A fluorescência observada dependia da intensidade da corrente ( $X$ ) e do tempo em que o tubo era submetido à corrente ( $Y$ ). Röntgen percebeu que dada a primeira seqüência, a próxima podia ser obtida descrevendo os números da seqüência anterior. Por exemplo: se a primeira seqüência for 2 então a próxima é 12 (ou seja, a seqüência anterior é formada por “um 2”), a seguinte 1112 (ou seja, a seqüência anterior é formada por um 1 e um 2), 3112 (ou seja, a seqüência anterior é formada por três 1 e um 2), e assim por diante.

Além de um cientista brilhante, Röntgen era extremamente organizado. Ele guardava todos os registros de seus experimentos. Infelizmente, com o tempo algumas seqüências foram danificadas e outras perdidas. Sua tarefa é dada uma seqüência, determinar as próximas  $K$  seqüências do experimento.

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

Cada instância é composta por uma linha contendo a primeira seqüência do experimento, formada por não mais de 1000 caracteres de 0 a 9, e o número  $K$  de seqüências que desejamos gerar ( $1 \leq K \leq 50$ ), respectivamente.

### Saída

Para cada instância, imprima a seqüência dada na entrada seguida de  $K$  linhas contendo as seqüências na ordem que foram geradas. As seqüências geradas não terão mais do que 2000000 caracteres.

Após cada instância imprima uma linha em branco.

## Exemplo de entrada

3  
2 5  
99 3  
000123 3

## Exemplo de saída

2  
12  
1112  
3112  
132112  
1113122112

99  
29  
1219  
11121119

000123  
30111213  
131031121113  
111311101321123113

## Problema B: Los buses de Cartagena

Arquivo: *marques*. [c/cpp/java]

Gabriel Garcia Marques é um escritor colombiano autor de histórias fantásticas como “Cién años de soledade”, “El amor en los tiempos del cólera” e “Memoria de mis putas tristes”. Suas histórias se caracterizam pelo uso do que ficou conhecido como “realismo mágico”, em que situações reais são explicadas com elementos mágicos. Apesar de seus trabalhos serem considerados muito ricos e até cenográficos, livros baseados em suas obras não têm merecido sucesso de público ou de crítica. O mais recente exemplo foi a filmagem em 2007 de “Love in the Time of Cholera”.

Uma de suas obras menos conhecidas é “Los buses de Cartagena”, que descreve a história de uma pequena companhia de ônibus da cidade colombiana que, principalmente devido aos problemas de quebra dos ônibus por excesso de carga, pretendia reduzir o número de passageiros transportados em cada viagem de Cartagena a Medellin para um mesmo número fixo. Ao mesmo tempo, a companhia queria continuar atendendo a todos os pedidos de forma satisfatória. Cada ônibus possui um horário de partida, e cada passageiro dispõe de uma lista de horários nos quais gostaria de viajar. Os passageiros desejam apenas ir para Medellin, ou seja, nenhum passageiro pretende viajar duas vezes no mesmo dia.

Sua tarefa é determinar o número mínimo de passageiros que devem ser transportados em cada viagem respeitando a restrição de que todos os passageiros devem ser atendidos.

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

A primeira linha de cada instância contém dois inteiros  $N$  e  $M$  ( $1 \leq N, M \leq 100$ ). Cada uma das  $M$  linhas seguintes possui o horário de partida de um dos ônibus. O horário está no formato  $hh:mm$  ( $00 \leq hh \leq 23$ ,  $00 \leq mm \leq 59$  e  $hh$  e  $mm$  possuem dois dígitos). Cada uma das  $N$  linhas seguintes contém a lista de horários em que cada passageiro pode viajar. A lista dos horários está no seguinte formato: um inteiro  $K$  ( $1 \leq K \leq M$ ) seguido de  $K$  horários, também no formato  $hh:mm$ , separados por um espaço em branco.

### Saída

Para cada instância imprima uma linha contendo o número mínimo de passageiros que devem ser transportados.

## Exemplo de entrada

3  
3 2  
00:10  
11:30  
1 00:10  
2 00:10 11:30  
2 11:30 00:10  
3 3  
23:50  
23:50  
23:51  
2 23:51 23:50  
1 23:50  
1 23:50  
4 2  
10:00  
12:01  
1 12:01  
1 12:01  
1 12:01  
1 12:01

## Exemplo de saída

2  
1  
4

## Problema C: Produto da guerra

Arquivo: *cruzverm.* [c/cpp/java]

O Comitê Internacional da Cruz Vermelha, organização sem fins lucrativos cujo objetivo é defender e amparar as vítimas de guerras (ou melhor, vítimas do capital) ou catástrofes naturais, ganhou os prêmios Nobel de 1917, 1945 e 1963 pelo seu importantíssimo trabalho. Como é de se imaginar, a Cruz Vermelha sempre teve problemas de locomoção no meio da guerra. Muitas ligações (estradas, ferrovias etc) entre cidades de países em guerra podem ser destruídas por bombardeios ou dominadas por tiranos.

O departamento de inteligência da Cruz Vermelha está empenhado em criar um programa de computador que auxilie as operações da Cruz Vermelha no futuro. A idéia é, dado um mapa da região que será ajudada, determinar em quais cidades devem ser feitas as bases da Cruz Vermelha. Inicialmente, o Departamento está interessado em testar a primeira versão do programa em cidades com as seguintes características: (a) sempre existe um caminho entre duas cidades que passa por uma ou mais ligações; (b) não existem dois caminhos diferentes entre duas cidades quaisquer. Apesar dos recursos da Cruz Vermelha geralmente serem limitados, eles querem escolher o maior número possível de bases, e garantir que ou existe uma base na cidade ou existe uma base em uma cidade vizinha, com a restrição adicional de que não é permitido criar bases em duas cidades vizinhas. Esta última restrição é dada pelo fato de que se estivesse em período de guerra, a Cruz Vermelha, como sabemos deve ter livre acesso nas cidades, e com isso pode surgir a suspeita de espionagem, o que pode comprometer o objetivo principal da organização.

Sua tarefa é escrever a primeira versão do programa que o Departamento quer testar.

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

A primeira linha de cada instância possui um inteiro  $N$  ( $1 \leq N \leq 600$ ) indicando o número de cidades do mapa. As cidades são identificadas por  $1, 2, \dots, N$ . As próximas  $N - 1$  linhas possuem dois inteiros  $u$  e  $v$  ( $1 \leq u, v \leq N, u \neq v$ ) que indicam uma ligação entre as cidades  $u$  e  $v$  (considere que tais ligações permitem acesso de  $u$  até  $v$  e de  $v$  até  $u$ ).

### Saída

Para cada instância imprima um inteiro indicando o número máximo de bases que a Cruz Vermelha consegue construir levando-se em consideração as restrições descritas anteriormente.

### Exemplo de entrada

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

### Exemplo de saída

7  
3

## Problema D: Viagens no tempo

Arquivo: *einstein*. [c/cpp/java]

Albert Einstein nasceu na Alemanha, mas foi na Suíça, trabalhando como funcionário público, que escreveu em 1905 os trabalhos que revolucionaram a Física moderna e o tornaram famoso. Em 1921 ganhou o prêmio Nobel de Física pela descoberta da lei do efeito fotoelétrico. Muitos acham seus trabalhos sobre a Teoria da Relatividade os mais importantes de sua carreira, entretanto não foram os que renderam o valioso prêmio.

Einstein gostava muito de fazer “experimentos mentais” para avaliar suas teorias. Um desses experimentos é muito famoso e descreve um elevador caindo com um relógio dentro. A idéia de viagens no tempo acabaram surgindo como possíveis, desde que se descobrisse como construir máquinas que pudessem viajar em velocidades maiores do que a velocidade da luz. Certamente, num futuro não muito distante, isso será possível e poderemos viajar livremente entre as eras e ver eventos como o descobrimento do Brasil em 1500, a chegada da Família Real em 1808 ou o Corinthians campeão da Libertadores em 2962 ao vivo.

Com as constantes viagens no tempo, será importante regular o serviço. As máquinas do tempo estarão espalhadas por toda a História e os viajantes terão de pegá-las para viajar para o presente ou para o futuro. Devido a restrições técnicas destas máquinas, não será possível viajar para qualquer instante do tempo diretamente, mas apenas para outros momentos históricos, de onde uma nova máquina poderá ser usada para seguir viagem. No entanto, estando em um momento histórico, você consegue ir para qualquer outro momento viajando por uma ou mais máquinas.

Juntamente com os viajantes do tempo, também surgirão os piratas da História, que tentarão roubar tesouros, inverter acontecimentos e mudar a história com os objetivos mais maldosos. Isso acarretará na criação da Polícia do Tempo. No ano de 2850 (antes do Corinthians ganhar sua primeira Libertadores) a Polícia do Tempo resolve isolar acontecimentos históricos, desabilitando ligações entre algumas máquinas. Cada ligação tem um custo associado para ser desabilitado, e sua tarefa é encontrar, dado um conjunto de momentos históricos, um conjunto de ligações – de custo mínimo – que ao serem desconectadas isolam os acontecimentos, ou seja, estando em uma máquina não será possível viajar para algumas das outras máquinas.

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

A primeira linha de cada instância contém dois inteiros  $N$  e  $M$  ( $1 \leq N \leq 100$  e  $1 \leq M \leq N(N-1)/2$ ) indicando o número de máquinas e o número de ligações, respectivamente. Cada uma das  $M$  linhas seguinte possui três inteiros  $u$ ,  $v$  e  $c$  ( $1 \leq u, v \leq N$ ,  $1 \leq c \leq 100$ ) que representam a existência de uma ligação entre a máquina  $u$  e  $v$  com custo  $c$ . Tal ligação pode ser usada para viajar da máquina  $u$  para máquina  $v$  e também da máquina  $v$  para máquina  $u$ .

### Saída

Para cada instância imprima uma linha contendo a soma dos custos das ligação que devem ser removidas.



### Exemplo de entrada

```
3
5 6
1 2 1
1 3 1
2 3 1
3 4 10
3 5 1
4 5 1
4 4
1 2 10
2 3 5
3 4 20
4 1 50
3 2
1 2 1
2 3 2
```

### Exemplo de saída

```
2
15
1
```

## Problema E: Equações diofantinas

Arquivo: *diofanto.[c/cpp/java]*

Diofanto de Alexandria viveu no terceiro século d.C. e é considerado por muitos o “pai da Álgebra”. Seu livro “Arithmetica” tratava da solução de equações algébricas com coeficientes inteiros para as quais se busca soluções também inteiras. Tais equações são conhecidas como equações diofantinas. Um grande estudioso do trabalho de Diofanto foi Pierre de Fermat, conhecido matemático francês.

Neste problema você deve resolver uma classe de equações diofantinas do tipo  $x_1 + x_2 + \dots + x_N = C$ . Ou seja, dados inteiros  $N$  e  $C$ , determine quantas soluções inteiras não-negativas existem para a equação  $x_1 + x_2 + \dots + x_N = C$ , onde  $0 \leq x_i \leq C$  para todo  $i = 1, 2, \dots, N$ .

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

Cada instância é composta por uma linha contendo dois inteiros  $N$  e  $C$  ( $1 \leq N, C \leq 1000000$ ).

### Saída

Para cada instância imprima uma linha contendo o número de soluções inteiras que respeitam as restrições.

### Exemplo de entrada

```
2
7 4
3 5
```

### Exemplo de saída

```
210
21
```

## Problema F: Crescimento das populações de bacilos

Arquivo: *koch*. [c/cpp/java]

Heinrich Hermann Robert Koch foi um médico alemão que viveu de 1843 a 1910 e ficou famoso por ter isolado o bacilo causador da tuberculose. Seus estudos sobre a doença que causava muitas mortes até meados do século XX possibilitaram o desenvolvimento de uma vacina que salvou milhões de vidas por todo o mundo. Robert Koch foi agraciado em 1905 com o prêmio Nobel de Medicina e é considerado um dos pais da Microbiologia.

Um dos estudos de Koch estava ligado com a velocidade de crescimento das populações de bacilos. Koch observou que os bacilos demoram um instante de tempo para atingir a maturidade e iniciar a divisão celular. A partir daí, o bacilo gera um novo indivíduo a cada instante de tempo por meio de uma divisão. Dessa forma, se partirmos de uma população inicial com apenas um indivíduo, no instante seguinte teremos ainda um (ele atinge a maturidade para divisão), no seguinte teremos 2, no outro 3, então 5 e assim por diante.

Sua tarefa é, dado um inteiro  $K$ , determinar os três últimos dígitos do número de bacilos após  $K$  instantes de tempo, partindo de uma população inicial com um indivíduo.

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

Cada instância é composta por apenas uma linha que contém um inteiro  $K$  ( $1 \leq K \leq 10^{1000000}$ ).

### Saída

Para cada instância imprima uma linha contendo os três últimos dígitos do número de bacilos após  $K$  instantes de tempo.

### Exemplo de entrada

```
5
1
4
10
21312
1000000
```

### Exemplo de saída

```
001
003
055
289
626
```

## Problema G: Elementar, meu caro Watson!

Arquivo: *wcw*. [c/cpp/java]

Watson, Crick e Wilkins receberam em 1962 o prêmio Nobel de Medicina especialmente pelo seu trabalho que resultou na descoberta da estrutura das moléculas do DNA e na sua importância na transmissão de informações entre as gerações de seres vivos. Watson e Crick publicaram na revista “Nature” em 1953 o artigo em que mostravam que a molécula de DNA apresentava uma estrutura de dupla hélice. O artigo assume enorme importância nos dias de hoje, especialmente depois dos vários avanços na área.

Muitas pesquisas têm sido feitas na área de Bioinformática ligadas à descoberta da seqüência de bases que compõem as moléculas de DNA dos vários seres vivos. Em especial, a estrutura destas moléculas tem sido usada para compor teorias de como os seres vivos evoluíram e quais têm ancestrais comuns. Acredita-se que os seres vivos presentes hoje no planeta podem descender de ancestrais comuns, sendo que as modificações nos seus respectivos DNAs são devidas a fenômenos de mutação ocorridos durante a evolução. Muitos biólogos acreditam no princípio da parcimônia, que diz que o número destas mutações deve ser o mínimo possível, uma vez que a Natureza busca, de certa forma, o caminho “mais barato” para a modificação desejada.

Sua tarefa neste problema é auxiliar os pesquisadores na tarefa de determinar se duas seqüências de DNA podem ter um ancestral comum. Considere dadas duas seqüências (podemos imaginar como seqüências de números inteiros). O seu objetivo é determinar o menor número de trocas de elementos de uma das seqüências (os elementos não precisam estar em posições adjacentes na seqüência) que leva uma das seqüências na outra. Observe que podemos considerar uma das seqüências fixa (por exemplo, em ordem crescente), dessa forma buscamos o número mínimo de tais trocas que ordena a seqüência dada.

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

A primeira linha de cada instância possui um inteiro  $N$  ( $1 \leq N \leq 10000$ ) indicando o número de inteiros na seqüência. A segunda linha contém uma permutação dos inteiros  $1, 2, \dots, N$  separados por espaço.

### Saída

Para cada instância imprima uma linha contendo o número mínimo de tais trocas que ordena a seqüência dada.

### Exemplo de entrada

```
2
5
2 3 4 5 1
5
2 1 4 5 3
```

### Exemplo de saída

```
4
3
```

## Problema H: Paz verde! Hipocrisia mundial!

Arquivo: *gore*. [c/cpp/java]

Albert Arnold Gore Jr é o nome completo do ex-vice-presidente dos Estados Unidos, Al Gore, ganhador do prêmio Nobel da Paz de 2007 pelo seu trabalho incessante de conscientizar a população mundial para as mudanças climáticas causadas pelo homem. O documentário “An Inconvenient Truth” vencedor do Oscar, mostra os efeitos causados pelo aquecimento global na paisagem do planeta e prevê um futuro catastrófico para a humanidade se a tendência de usurpar os recursos do planeta não for mudada.

Al Gore cresceu em Washington DC uma vez que seu pai foi deputado e depois senador pelo Tennessee. Graduou-se em Harvard em 1969 e foi um ativista contra a guerra do Vietnam e chegou a apoiar o líder Martin Luther King na sua luta contra a segregação racial. Sua atuação como vice-presidente dos Estados Unidos na administração de Bill Clinton também foi excepcional. Apesar de ter tido mais votos que o concorrente do partido republicano, perdeu as eleições presidenciais e afastou-se da disputa da presidência.

Um dos seus trabalhos mais importantes diz respeito ao posicionamento ótimo de fornos em produção de tijolos. O processo de fabricação de tijolos é bastante poluente, e exige a queima em alta temperatura do barro a fim de que o tijolo atinja a consistência desejada. A queima consome grandes quantidades de madeira, produzida em fazendas para este fim. Estudos da Universidade de Harvard mostram que há uma distância máxima para o posicionamento nesses fornos: se estiverem muito distantes, a dispersão do calor não permite que a queima seja feita por igual, trazendo prejuízos à produção de tijolos e também ao meio ambiente. Uma vez que os fornos são posicionados no meio da floresta (que é cortada para a queima), as distâncias são medidas usando a métrica de Manhattan, ou seja, a distância entre dois pontos é dada pela soma dos valores absolutos das diferenças das coordenadas. Sua tarefa é, dada a localização de vários fornos numa fazenda, e uma distância  $D$ , determinar, para cada um dos fornos, quantos fornos estão à distância no máximo  $D$ . Com estes dados será possível determinar quais fornos precisam ser acesos simultaneamente sem prejuízos econômico ou ambiental.

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

A primeira linha de cada instância possui dois inteiros  $N$  e  $D$  ( $1 \leq N, D \leq 100000$ ) representando o número de fornos e uma distância, respectivamente. Cada uma das próximas  $N$  linhas possui dois inteiros  $x$  e  $y$  ( $0 \leq x, y \leq 100000$ ) que indicam a posição de um forno.

### Saída

Para cada instância imprima uma linha contendo  $N$  inteiros que indicam quantos fornos estão à distância no máximo  $D$  dos fornos  $1, 2, \dots, N$ .

### Exemplo de entrada

1  
13 2  
0 2  
1 3  
1 2  
1 1  
2 4  
2 3  
2 2  
2 1  
2 0  
3 3  
3 2  
3 1  
4 2

### Exemplo de saída

4 7 7 7 4 7 12 7 4 7 7 7 4

## Problema I: Sonhos, acredite neles!

Arquivo: *king*. [c/cpp/java]

Um dos mais importantes ativistas políticos do mundo foi o Dr. Martin Luther King Jr, cujo discurso mais conhecido foi “I have a dream”. Em 1964, ele recebeu o Nobel da Paz por seu empenho na luta pelo fim do preconceito racial nos Estados Unidos, e pela sua liderança nos movimentos não violentos. Pouco tempo depois de ter recebido o prêmio, Luther King foi assassinado momentos antes de uma marcha no Memphis.

Além do empenho na luta política, Luther King gostava de jogar quebra-cabeça. Um dos jogos que ele adorava jogar é o seguinte: são dados dois mapas  $N$ -por- $M$ , cada um com um robô. Cada mapa contém um ponto inicial e um final. Algumas “casas” do mapa são cercadas por paredes. Uma casa do mapa pode ser ou não um buraco. Um comando dado (Cima, Baixo, Esquerda, Direita) é executado ao mesmo tempo para ambos os mapas. Os robôs não atravessam as paredes e nem flutuam sobre os buracos. O objetivo é chegar com os dois robôs no ponto final ao mesmo tempo, em até 50 movimentos, se isso for possível.

Neste problema, sua tarefa é dados dois mapas  $N$ -por- $M$ , determinar o número mínimo de movimentos que resolve o problema.

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

A primeira linha da instância possui dois inteiros  $N$  e  $M$  ( $1 \leq N, M \leq 50$ ), indicando o número de linhas dos mapas e o número de colunas dos mapas, respectivamente. Nas linhas seguintes são dados os dois mapas. Para cada mapa teremos  $N$  linhas com  $M$  caracteres. O caractere “.” indica uma posição livre; “#” indica uma posição cercada por paredes; “B” indica um buraco; “R” indica a posição inicial do robô e “F” indica a posição final do robô.

### Saída

Para cada instância imprima uma linha contendo o número mínimo de movimentos que resolve o problema, ou `impossivel` se não for possível resolver o problema com no máximo 50 movimentos.

## Exemplo de entrada

```
2
4 4
....
....
...F
..#R
....
....
.FBB
#..R
4 4
.BFB
...#
.#BB
...R
####
.BBF
....
#R..
```

## Exemplo de saída

```
3
12
```



## Problema J: Alocação ótima de commodities

Arquivo: *koopmans*. [c/cpp/java]

Tjalling C. Koopmans ganhou em 1975 o prêmio Nobel de Economia juntamente com o matemático russo Kantorovich pelas suas contribuições em importantes áreas como a alocação ótima de recursos. Koopmans formou-se em Matemática pela Universidade de Utrecht, na Holanda, e se especializou em economia matemática. Durante a segunda guerra mundial esteve envolvido no estudo de alocação ótima de recursos, que 30 anos mais tarde lhe rendeu o prêmio Nobel. É considerado um dos precursores da teoria de programação linear. Suas contribuições têm importantes aplicações em Economia, Matemática, Física e mesmo em Química.

Um dos problemas prediletos de Koopmans era o de alocação ótima de commodities. Neste problema, é dado um valor inicial e um valor final da aplicação a ser feita. Entretanto, nem todos os valores podem ser aplicados nos vários investimentos. Cada investimento é definido através de um número inteiro, e, por convenção, apenas quando o valor a ser aplicado for um múltiplo de pelo menos um número que define um investimento ele pode ser aplicado.

Sua tarefa neste problema é calcular o valor máximo que pode ser aplicado. Ou seja, dado o valor inicial e valor final a serem aplicados e uma lista de inteiros que definem as várias aplicações, você deverá calcular a soma dos valores que podem ser aplicados no intervalo.

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

A primeira linha de cada instância possui três inteiros  $I, F$  e  $N$  ( $1 \leq I \leq F \leq 1000000000$  e  $1 \leq N \leq 20$ ) que representam o valor inicial, o valor final e o número de elementos da lista de aplicações. A próxima linha contém  $N$  inteiros  $1 \leq a_i \leq 1000000000$  indicando a lista de aplicações.

### Saída

Para cada instância imprima uma linha contendo a soma dos valores que podem ser aplicados no intervalo. Como este valor pode ser muito grande então imprima o resultado módulo 1300031.

### Exemplo de entrada

```
3
1 10 1
1
1 9 2
3
5
1 999 2
3
5
```

### Exemplo de saída

```
55
23
233168
```

## Problema K: O Fantástico Jaspion!

Arquivo: *jaspion*. [c/cpp/java]

Em 1985 estréia na TV Japonesa a série Kyojiu Tokusou Jaspion (Investigador Especial de Monstros Jaspion). A série chega ao Brasil alguns anos depois com o título “O Fantástico Jaspion”, e com ela nasce a fantasia de polícia espacial em milhões de brasileirinhos. As crianças saíam da escola, corriam pelas ruas (sem olhar se vinha carro), ligavam a TV e mergulhavam na coragem, exemplo de pessoa, e incontestável sede por justiça do Fantástico Jaspion. O comércio de gibis e as brigas por figurinhas no recreio da escola estavam alcançando números históricos. Até então, tal sentimento só havia sido estimulado com tanta intensidade pelo Chaves e a sua turma! Diante dessa febre inter-galática, o inevitável aconteceu. Os produtores do Jaspion ganharam o Nobel da Paz! Isso mesmo! Os produtores ganharam um Nobel. As histórias do grandioso Jaspion estavam por todo canto. Agora as crianças tinham um belíssimo exemplo para seguir. A paz mundial estava garantida! Não precisávamos mais temer o monstrengo Satan Gos!

No Brasil havia uma criança que adorava as histórias do Jaspion! Antônio Melhorança Capote Valente Junior carinhosamente apelidado de ACM, um menino da zona sul de São Paulo que adorava cantar as músicas do grande herói. Ele era tão fanático que chegou a comprar um dicionário de Japonês-Português e iniciou um trabalho árduo de tradução. Entretanto, o trabalho ficou inacabado! Alguns trechos da canção ainda precisam ser traduzidos. Neste momento você deve estar se perguntando: qual é a minha tarefa neste fabuloso problema? Ok! Antes de falar da sua tarefa, convide seu companheiro de equipe para mergulhar com você no desfecho da história. Para isso, vamos falar mais um pouco sobre o nosso ACM. Ele se formou em Ciência da Computação e hoje trabalha no mesmo escritório que você. Pois é! Você trabalha como programador ao lado dessa figura! Como sabemos que você gosta muito dele, temos certeza que vai aceitar a seguinte tarefa: dado um dicionário Japonês-Português e uma letra de música, escreva um programa que imprima a letra traduzida.

### Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro  $T$  indicando o número de instâncias.

A primeira linha de cada instância contém dois inteiros  $M$  e  $N$  ( $1 \leq M \leq 1000000, 1 \leq N \leq 1000$ ), que representam o número de palavras no dicionário e o número de linhas na letra da música, respectivamente.

Os próximos  $M$  pares de linhas contêm as traduções: a primeira linha de cada par contém a palavra em Japonês, e a segunda linha contém a tradução para o Português (que pode ter uma ou mais palavras). Todas as palavras usam apenas letras minúsculas. Cada palavra em Japonês aparece apenas uma vez em cada instância.

As próximas  $N$  linhas contêm a letra da música. Cada linha da letra da música é uma lista de palavras separadas por um espaço (todas as palavras consistem apenas de letras minúsculas). Algumas podem estar vazias, mas nenhuma linha possui espaços no início ou no final.

Nenhuma linha contém mais do que 80 letras.

### Saída

Para cada instância imprima as  $N$  linhas traduzidas. As palavras que não estão no dicionário devem ser impressas como aparecem na entrada. Imprima uma linha em branco no final de cada tradução.

Nenhuma linha da saída contém mais do que 80 letras.

## Exemplo de entrada

1  
4 3  
galaxy  
cara tossiu  
kagayaku  
canalha do  
atsuki  
alto que  
yuushi  
util  
o galaxy  
o galaxy  
o kagayaku atsuki yuushi

## Exemplo de saída

o cara tossiu  
o cara tossiu  
o canalha do alto que util