102 – A Mochila Binária Ecológica – 86,95%

Conhecimento Geral

O problema da mochila binária (ou empacotamento binário), ou a colocação de objetos de certos pesos em diferentes locais, sujeitos a algumas restrições são problemas de interesse histórico. Alguns dos problemas de empacotamento são NP-completos, mas são amenizados por soluções de programação dinâmica ou por soluções heurísticas que garantem uma proximidade com a solução ótima.
Neste problema, você irá resolver um problema de empacotamento que lida com reciclagem de garrafas de vidro.

O Problema

A reciclagem de vidro exige que o vidro seja separado de acordo com sua cor em uma das três categorias: marrom (B-*brown*), verde (G-*green*) e transparente (C-*clean*). Neste problema você receberá três contêineres de reciclagem, cada um contendo um número específico de garrafas marrons, verdes e transparentes. Para poderem ser recicladas, as garrafas precisarão ser movidas de forma que cada contêiner fique com garrafas de apenas uma cor.
O problema consiste em minimizar o número de garrafas que serão movidas. Você pode pressupor que o único problema é minimizar o número de movimentos entre contêiner.
Para este problema, considere que cada contêiner tem capacidade infinita e a única restrição é que você deverá mover as garrafas de forma a cada contêiner ficar com garrafas de uma única cor. O número total de garrafas nunca será maior que 2^{31} (2.147.483.648).

A Entrada

A entrada consiste de uma série de linhas, cada uma contendo nove números inteiros. Os três primeiros números representam o número de garrafas marrons, verdes e transparentes (respectivamente) do contêiner número 1, os três inteiros seguintes correspondem, respectivamente, ao número de garrafas marrons, verdes e transparentes do contêiner número 2 e os três últimos números correspondem, respectivamente, ao número de garrafas marrons, verdes e transparentes do contêiner número 3.
Por exemplo, a linha: **10 15 20 30 12 8 15 8 31** indica que há 20 garrafas transparentes no contêiner 1, 12 garrafas verdes no contêiner 2 e 15 garrafas marrons no contêiner 3.
Os números inteiros em uma linha serão separados por um ou mais espaços. Seu programa deverá processar todas as linhas do arquivo de entrada.

A Saída

Para cada linha de entrada haverá uma linha de saída indicando que cor de garrafas irá para cada contêiner de forma a minimizar o número de movimentos de garrafas. Você também deverá imprimir o número mínimo de movimentos.

A saída consistirá de uma *string* de três caracteres maiúsculos 'G', 'B', 'C' (representando as cores verde, marrom e transparente) e corresponderão as cores associadas a cada contêiner.

O primeiro caractere dessa *string* representará a cor associada ao primeiro contêiner, o segundo caractere representará a cor associada ao segundo contêiner e o terceiro caractere corresponderá à cor associada ao terceiro contêiner.

O inteiro indicando o número mínimo de movimentos deverá aparecer após a *string*.

Se mais de uma ordem de contêineres marrom, verde e transparente atingir o número mínimo de movimentos, então a representação com menor "ordem alfabética" deverá ser apresentada (por exemplo CBG tem menor ordem alfabética do que GBC).

Exemplo de Entrada

```
1 2 3 4 5 6 7 8 9
5 10 5 20 10 5 10 20 10
```

Exemplo de Saída

```
BCG 30
CBG 50
```

1160 - Explosivos – 91,62%

Um serviço secreto desenvolveu um novo tipo de explosivo que ativa sua propriedade volátil apenas quando uma associação específica de produtos ocorre. Cada produto é uma mistura de dois componentes simples, a qual nós chamaremos de *par de ligação*. Se N>2, então ao se misturar N diferentes *pares de ligação* usando N componentes simples cria-se um poderoso explosivo. Por exemplo, os *pares de ligação*: A+B, B+C, A+C (três pares formados a partir de três componentes) resultam em um explosivo, enquanto A+B, B+C, A+D (três pares, quatro componentes) não.

Você não é um agente secreto e sim apenas um cara de uma empresa de entregas com um problema perigoso: receber *pares de ligação* em ordem sequencial e então colocá-los em um navio de carga. No entanto, você deve evitar colocar no mesmo cômodo uma associação explosiva. Então, após colocar um conjunto de pares, se você receber um par que pode produzir uma explosão com alguns dos outros pares já colocados lá, você deve recusá-lo; caso contrário, você deve aceitá-lo. Por exemplo: Vamos assumir que você está recebendo a seguinte sequência: A+B, G+B, D+F, A+E, E+G, F+H. Você deverá aceitar os primeiros quatro pares, mas então recusar E+G, já que isso possibilitaria a criação do seguinte explosivo: A+B, G+B, A+E, E+G (quatro pares com quatro componentes).

Finalmente, você poderá aceitar o último par, F+H. Escreva um programa que compute o número de pares recusados dada uma sequência de pares de ligação.

Entrada

A entrada conterá **vários casos de teste**, cada um deles descritos da seguinte maneira. Casos de teste consecutivos são separados por uma linha em branco.

Ao invés de letras, nós usaremos números inteiros para representar componentes. A entrada contém várias linhas. Cada linha (exceto a última) consiste de dois inteiros (cada inteiro valerá entre 0 e 10^5) separados por um espaço em branco, representando um par de ligação. Cada caso de teste termina com uma linha contendo o número “-1”. Você pode assumir que não haverá a repetição de um mesmo *par de ligação* em uma única entrada.

Saída

Para cada caso de teste, a saída deve ser da seguinte forma: uma única linha com o número de recusas.

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

1174 - IP-TV – 92,62%

Um consórcio europeu de provedores de Internet gerencia um *backbone* de internet, com links diretos (conexões) entre um grande número de cidades europeias. Um link entre duas cidades é bidirecional. A transmissão de uma mensagem em um link tem um custo associado. Como é comum na Internet, é possível usar uma sequência (ilimitada) de links diretos para transmitir indiretamente dados entre qualquer par de cidades.

Para possibilitar a transmissão em broadcast de programas de TV usando este backbone, é necessário enviar continuamente dados para todos os nós da rede. A fim de ajudar a minimizar os custos, é necessário selecionar os links da rede que serão usados para transmitir os dados. O conjunto de links selecionados deve ser conexo e incluir todos os nós na rede. Para auxiliar este consórcio a gerenciar sua rede, você foi chamado para criar um programa que compute o custo mínimo para transmitir dados para todas as cidades do backbone.

Dado um conjunto de links de rede, compute o custo mínimo de transmissão para atingir todos os nós.

Entrada

A entrada consiste de múltiplos casos de teste, a primeira linha da entrada contém o número de casos de teste.

Haverá uma linha em branco antes de cada caso de teste. A primeira linha de cada caso de teste contém um inteiro positivo M, menor ou igual a 2.000, indicando o número de cidades que têm conexão na rede. A segunda linha contém um inteiro N menor ou igual a 50.000, com o número de links. Cada uma das N linhas seguintes contém a representação de um link.

Cada linha contém duas strings e um inteiro, separados por espaços em branco, B E C, sendo B e E o nome das cidades (com no máximo 8 caracteres cada) e C um inteiro positivo menor ou igual a 30, representando o custo de transmissão do link.

Saída

Para cada caso de teste, a saída consistirá de uma linha contendo um inteiro com o custo mínimo de transmissão para se enviar dados para todas as cidades. Imprima uma linha em branco entre casos de teste (mas não após o último caso de teste).

Exemplo de Entrada	**Exemplo de Saída**
1 4 5 lisbon london 6 lisbon paris 5 london paris 1 london berlin 2 paris berlin 10	8

1237 - Esperto o bastante? – 92,60%

A companhia *Auto-mobile Charting & Manufacturing* (ACM) é uma companhia especializada na fabricação de partes de automóveis. Sendo uma das líderes mundiais no setor, ACM certamente quer manter-se atualizada no mundo automobilístico. Em seu aniversário de 100 anos, ACM compilou uma enorme lista de variações de preços de todos os automóveis na história. ACM deseja desenvolver um programa que eles chamam de *Automobile Expert System* (AES). O programa recebe o preço P como entrada e procura por uma montadora para a qual P fica dentro da faixa L (menor preço) e H (maior preço) dos carros feitos por essa montadora. Se não houver **exatamente** uma montadora que satisfaça a consulta, seu programa deverá imprimir a saída "UNDETERMINED". Esse programa nem parece tão esperto, né? Você deverá desenvolver esse programa para ACM.

Entrada

A entrada começa com uma linha contendo um inteiro T (T<=10), correspondendo ao número de casos de teste. Cada caso de teste se inicia com o tamanho do banco de dados D (D <10000). Cada uma das D linhas seguintes contém M, L e H ($0 < L < H < 1000000$) que são os nomes das companhias (sem espaços em branco e com no máximo 20 caracteres), o preço do carro mais barato já fabricado pela companhia (L), e o preço do carro mais caro já fabricado pela companhia (H). Então haverá um número de consultas Q (Q<1000), seguidas por Q linhas contendo P ($0 < P < 1000000$), que é o preço utilizado na consulta.

Saída

Para cada consulta você deverá imprimir uma linha contendo o nome da montadora ou da string "UNDETERMINED" se nenhuma montadora ou mais de uma montadora satisfizer a consulta.

É necessário imprimir uma linha em branco entre casos de teste (mas não após o último).

Exemplo de Entrada

```
1
4
HONDA 10000 45000
PEUGEOT 12000 44000
BMW 30000 75900
CHEVROLET 7000 37000
4
60000
7500
5000
10000
```

Exemplo de Saída

```
BMW
CHEVROLET
UNDETERMINED
UNDETERMINED
```

1368 - Consenso de DNA – 94,62%

O DNA (ácido desoxirribonucleico) é uma molécula que contém as instruções genéticas. Ele é composto de quatro diferentes nucleotídeos, chamados Adenina, Timina, Guanina e Citosina, conforme mostrado na figura 1. Se representarmos um nucleotídeo por seu caractere inicial, uma sequência de DNA pode ser representada com uma longa string (uma sequência de caracteres) composta por quatro caracteres: A, T, G e C. Por exemplo, assuma que nós recebemos um pedaço de uma sequência DNA que é composta pelos seguintes nucleotídeos: "*Thymine-Adenine-Adenine-Cytosine-Thymine-Guanine-Cytosine-Cytosine-Guanine-Adenine-Thymine*"

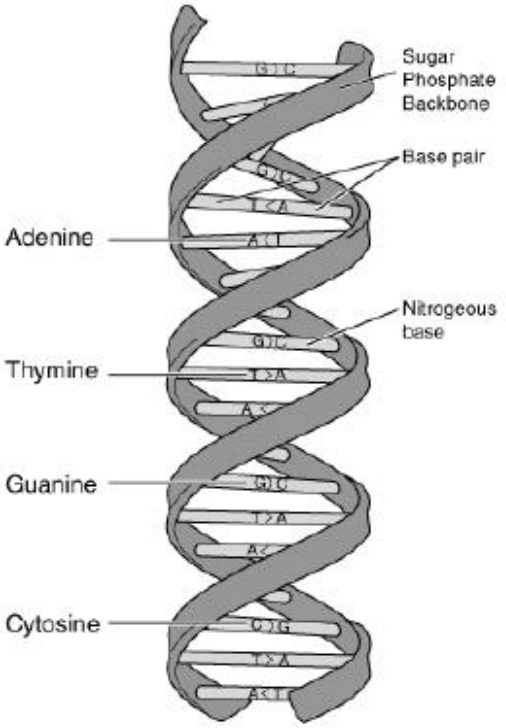

Figura 1

Nós podemos representar essa sequência com a string "TAACTGCCGAT". Um biólogo descobriu que o gene X comumente existe na sequência de DNA de cinco diferentes tipos de animais: cães, gatos, cavalos, vacas e macados. Ele também descobriu que as sequências de DNA do gene X são muito parecidas (ver figura 2).

Cat:	GCATATGGCTGTGCA
Dog:	GCAAATGGCTGTGCA
Horse:	GCTAATGGGTGTCCA
Cow:	GCAAATGGCTGTGCA
Monkey:	GCAAATCGGTGAGCA

Figura 2. Sequências de DNA para o gene X de cinco organismos.

O biólogo pensa que os seres humanos também podem ter o gene X e decidiu pesquisar pela sequência de DNA do gene X no DNA humano. No entanto, antes de procurar, ele deverá definir uma sequência representativa do gene X porque suas sequências não são exatamente as mesmas nos cinco animais. Ele decidiu usar a distância Hamming para definir a sequência representativa. A distância Hamming é o número de caracteres diferentes em cada posição de duas strings de mesmo tamanho. Por exemplo, considere as seguintes sequências: "AGCAT" e "GGAAT". A distância Hamming dessas duas strings é 2 porque os caracteres na 1ª e na 3ª posição dessas strings são diferentes. Usando a distância Hamming podemos definir a string representativa para um conjunto de múltiplas strings de tamanho igual. Dado um conjunto de strings $S = s_1,..., s_m$ de tamanho n, o erro de consenso entre a string y de tamanho n e o conjunto S é dado pela soma das distâncias Hamming entre y e cada uma das sequências em S. Se o erro de consenso entre y e S é o menor entre

todas as strings de tamanho n, y é chamado de string consenso de S. Por exemplo, dadas as três strings "AGCAT", "AGACT" e "GGAAT" a string consenso é "AGAAT" porque a soma da distância Hamming entre "AGAAT" e as três strings é 3, e nenhuma outra string de tamanho 5 terá distância menor (neste caso, a string consenso é única, mas em geral poderá haver mais de uma sequência consenso). Nós usamos a sequência consenso como a sequência representativa de DNA. Por exemplo, a string consenso do gene X apresentado na figura 2 é "GCAAATGGCTGTGCA" e o erro de consenso é 7.

Entrada

A entrada consistirá de T casos de teste. O número de casos de teste T é dado na primeira linha de entrada. Cada caso de teste inicia com uma linha contendo dois inteiros m e n, separados por um espaço em branco. O inteiro m (4<=m<=50) representa o número de sequências de DNA e n (4<=n<=1000) representa o tamanho das sequências de DNA. Em cada uma das m linhas seguintes, cada uma das sequências de DNA será apresentada.

Saída

Seu programa deverá imprimir para cada caso de teste duas linhas. Na primeira deverá imprimir a sequência consenso e na segunda o valor do erro de consenso. Se há mais de uma sequência consenso, imprima a lexicograficamente menor.

Exemplo de Entrada

```
3
5 8
TATGATAC
TAAGCTAC
AAAGATCC
TGAGATAC
TAAGATGT
4 10
ACGTACGTAC
CCGTACGTAG
GCGTACGTAT
TCGTACGTAA
6 10
ATGTTACCAT
AAGTTACGAT
AACAAAGCAA
AAGTTACCTT
AAGTTACCAA
TACTTACCAA
```

Exemplo de Saída

```
TAAGATAC
7
ACGTACGTAA
6
AAGTTACCAA
12
```

1583 - Gerador de Dígitos – 95,31%

Para um dado inteiro positivo N, a soma de dígitos de N é definida como a soma do valor N e de seus dígitos.
Quando M é a soma de dígitos de N, dizemos que N é o gerador de M.
Por exemplo, a soma de dígitos de 245 é 256 (= 245 + 2 + 4 + 5). Assim, 245 é o gerador de 256.
Não é surpresa que alguns números não têm nenhum gerador e alguns têm mais de um gerador. Por exemplo, os geradores de 216 são 198 e 207.
Escreva um programa que encontre o menor gerador de um dado inteiro.

Entrada

A entrada consiste de T casos de teste. O número de casos de teste T é dado na primeira linha da entrada. Cada caso de teste é apresentado em uma linha e contém um inteiro N (1<=N<=100000).

Saída

Seu programa deverá imprimir uma linha para cada caso de teste. A linha deverá conter o menor gerador de N, para cada caso de teste. Se N não possui nenhum gerador, imprima 0 (zero).

Exemplo de Entrada

```
3
216
121
2005
```

Exemplo de Saída

```
198
0
1979
```