6^a Seletiva da UFPR

4 de Agosto de 2015

Flávio Zavan
Ricardo Oliveira

Instruções Importantes

- Em cada problema, cada arquivo de entrada contém apenas um caso de teste. Sua solução será executada com vários arquivos de entrada.
- Se a solução der erro ou esgotar o tempo limite para um dado arquivo de entrada, você receberá a indicação de erro (estouro de tempo, resposta errada, etc.) para aquele arquivo, e a execução terminará. O arquivo que causou o erro não é identificado. Note que pode haver outros erros, de outros tipos, para outros arquivos de entrada, mas apenas o primeiro erro encontrado é reportado.
- Sua solução será compilada com a seguinte linha de comando:
  - C: `gcc -static -O2 -lm`
  - C++: `g++ -static -O2 -lm`
  - C++11: `g++ -std=c++11 -static -O2 -lm`
  - Java: `javac`
  - Pascal: `fpc -Xt -XS -O2`
- Sua solução deve processar cada arquivo de entrada no tempo máximo estipulado para cada problema, dado pela seguinte tabela:

Problema	**Nome**	**Tempo Limite (segundos)**
A	R mais L é igual a J	1
B	Pedágio da Nlogônia	2
C	Ingress	1
D	Novo Recorde	1
E	Final de Fisiologia Canina	3
F	Cardápio do RUFNl	2
G	Balas de Morango	1
H	Robô Aspirador	1
I	Circo de Pulgas	1
J	Vírus Nlogono	1

- Os juizes usam um sistema de 64 bits (idêntico às maquinas do DINF).
- Todas as linhas, tanto na entrada quanto na saída, terminam com o caractere de fim-de-linha ($\backslash n$), mesmo quando houver apenas uma única linha no arquivo.
- Para submissões em **JAVA**, a classe deverá ter o mesmo nome que o *basename* do problema (leia a linha entre o título e o texto do problema).

A: R mais L é igual a J

`Arquivo: rlj.[c|cpp|java|pas]`

Durante sua grande aventura na Terra do Oeste, Joãozinho descobriu um livro sagrado que, segundo as lendas, foi escrito pelos próprios deuses antigos. Uma passagem em particular chamou a atenção do jovem aventureiro:

"A origem daquele que nada sabe se revelará quando aquele escolhido pelos deuses desvendar o enigma por eles lhe imposto. R+L=J."

O enigma o intrigou bastante. Joãozinho logo começou a procurar por valores de R, L e J que satisfazem a equação citada na passagem. Após investigações, o jovem encontrou dois dos três valores citados. Joãozinho deve agora determinar o terceiro dos valores citados, para que o enigma seja solucionado e para que *"a origem daquele que nada sabe"* seja revelada.

Entrada

A única linha da entrada contém uma *string* na forma `R+L=J`. Se uma variável tem um valor conhecido, tal valor aparece na *string* no lugar da variável. Caso contrário, a letra que representa a variável aparece normalmente.

É garantido que exatamente dois dos três valores são conhecidos. Além disso, todos os valores *conhecidos* estão entre 1 e 10^6, inclusive. Não há zeros à esquerda nos valores dados.

Saída

Imprima uma linha contendo o valor da variável desconhecida.

Exemplo de entrada	**Exemplo de saída**
`3+8=J`	`11`

Exemplo de entrada	**Exemplo de saída**
`5+L=5`	`0`

Exemplo de entrada	**Exemplo de saída**
`R+7=5`	`-2`

B: Pedágio da Nlogônia

`Arquivo: pedagio.[c|cpp|java|pas]`

Bem-vindo a Nlogônia! Há N cidades na Nlogônia, e algumas delas são conectadas por rodovias bidirecionais (como usual). Neste país, há exatamente um caminho entre qualquer par de cidades.

Para atravessar uma rodovia (em qualquer uma das duas direções), um pedágio deve ser pago. O valor do pedágio de cada rodovia é conhecido. O ministério do turismo da Nlogônia está levantando um estudo sobre o sistema de pedágios em seu país. Para auxiliar o ministério, você deverá responder a Q consultas. Em cada consulta, são dadas duas cidades A e B e um inteiro K. Sua tarefa é determinar os valores dos K maiores pedágios presentes no caminho da cidade A para a cidade B.

Entrada

A primeira linha contém o inteiro N ($2 \leq N \leq 10^5$). As cidades são numeradas de 1 a N. As próximas $N-1$ linhas descrevem as rodovias. Cada uma contém três inteiros C_i, C_j e P ($1 \leq C_i, C_j \leq N, C_i \neq C_j, T \leq 10^9$), indicando uma rodovia que conecta as cidades C_i e C_j e cujo pedágio custa P centavos nlogonenses.

A próxima linha contém o inteiro Q ($1 \leq Q \leq 10^5$), o número de consultas. As próximas Q linhas descrevem as consultas. Cada uma contém três inteiros A, B e K ($1 \leq A, B \leq N, A \neq B, 1 \leq K \leq 5$).

Saída

Para cada consulta, imprima os K maiores pedágios no caminho de A para B. Imprima os valores em ordem não crescente, separados por um espaço. Se há menos de K rodovias no caminho de A a B, imprima o valor do pedágio de todas elas.

Exemplo de entrada	**Exemplo de saída**
6 1 2 1 1 3 5 3 4 3 3 5 8 1 6 2 3 2 6 3 5 2 1 4 6 2	2 1 8 5 3

C: Ingress

`Arquivo: ingress.[c|cpp|java|pas]`

Ingress é um jogo de realidade aumentada bastante jogado no mundo inteiro. O jogo pode ser descrito como uma gincana mundial. Duas facções disputam entre si para ter o controle da maior área no mundo.

O mapa do jogo pode ser descrito como um plano. Há N *portais* no mapa. Um portal i pode ser descrito como um ponto (x_i, y_i) no plano.

Para cada par de portais distintos, um jogador pode fazer um *link* entre ambos. Quando um *link* entre dois portais é criado, um segmento de reta ligando os portais é desenhado no mapa. Em um conjunto de três portais distintos, quando há um *link* entre todos os pares de portais, um *control field* é criado. Um *control field* é definido pelo triângulo formado pelos *links* feitos entre os portais. O tamanho de um *control field* é dado pela área deste triângulo.

Pelas regras do jogo, dois *control fields* distintos não podem possuir intersecção positiva. Desta forma, dois *control fields* podem, no máximo, compartilhar um *link* ou um portal.

Sua missão é, dadas as posições de todos os portais, determinar a maior soma possível dos tamanhos dos *control fields* que podem ser criados.

Entrada

A primeira linha contém um inteiro N ($3 \leq N \leq 10^5$), o número de portais. As próximas N linhas contém dois inteiros x_i e y_i cada ($0 \leq x_i, y_i \leq 10^4$), indicando as coordenadas dos portais no mapa.

Saída

Imprima uma linha contendo a maior soma possível dos tamanhos dos *control fields* que podem ser criados. Arredonde e imprima a resposta com exatamente duas casas decimais.

Exemplo de entrada	**Exemplo de saída**
3 1 1 5 1 3 3	4.00

Exemplo de entrada	**Exemplo de saída**
4 1 2 3 5 1 5 3 2	6.00

D: Novo Recorde

`Arquivo: recorde.[c|cpp|java|pas]`

A grande Maratona de Rua de Curitiba irá ocorrer nos próximos dias! Vários atletas estão treinando há dias para o grande dia da corrida. Flávio é um dos atletas que está treinando diariamente para se sair bem na corrida. Ele tem corrido todas as manhãs nas pistas próximas de sua casa.

Os treinos do garoto são monitorados por um aplicativo em seu celular. Após cada treino, Flávio sabe tanto a duração do treino quanto a distância total percorrida. Com essas informações, ele consegue determinar a velocidade média obtida em cada treino.

Flávio está muito preocupado com a evolução de seu desempenho nos treinos, e em particular com seu recorde de velocidade média. Tal recorde é batido em um dado treino quando a velocidade média para este treino é maior que todas as velocidades médias obtidas nos treinos anteriores. Ajude Flávio a determinar em quais treinos ele conseguiu bater seu recorde.

Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 30$), o número de treinos feitos. Considere que os treinos foram feitos nos dias $1, 2, ..., N$. As próximas N linhas descrevem os treinos. A linha i ($1 \leq i \leq N$) contém dois inteiros T_i e D_i ($1 \leq T_i, D_i \leq 100$), indicando, respectivamente, a duração do treino (em minutos) e a distância percorrida no treino (em quilômetro).

Saída

Imprima uma lista de inteiros indicando os dias nos quais o recorde foi batido. Cada dia deve ser impresso em uma linha. Imprima os dias em ordem crescente. Note que o dia 1 sempre deve ser impresso.

Exemplo de entrada	**Exemplo de saída**
3 1 1 2 1 2 3	1 3

Exemplo de entrada	**Exemplo de saída**
2 2 16 4 20	1

E: Final de Fisiologia Canina

`Arquivo: final.[c|cpp|java|pas]`

Vasya está frito. Passou o semestre inteiro indo pescar e andar de bicicleta e ficou para fazer a prova final da disciplina de Fisiologia Canina.

Após receber a folha de questões da professora, Vasya notou que há apenas uma única questão. Existem duas colunas, a da esquerda, com N linhas, é uma lista de raças de cães e a da direita, com M linhas, uma lista de processos fisiológicos.

Espera-se que o aluno relacione as raças a seus processos fisiológicos exclusivos. Desta forma, podem ocorrer casos tanto de raças quanto de processos sem pares. Entretanto, é nenhuma raça nem processo deverá ser relacionado com mais de um item da outra coluna.

Desesperado, Vasya anotou no canto da página, para cada raça, quais processos fisiológicos ele acredita poder estarem relacionados. Buscando se acalmar, ele espera determinar o número máximo de pares os quais ele consegue ligar dentro das possibilidades anotadas.

Entrada

A primeira linhas contém dois inteiros N e M ($1 \leq N, M \leq 10^5$), o número de raças e o número de processos fisiológicos, respectivamente. As próximas N linhas contém um inteiro e_i ($0 \leq e_i \leq M$), a quantidade de processos fisiológicos que podem ser relacionaos à i-ésima raça, seguido de e_i inteiros, os processos, numerados de 1 a M, possivelmente relacionados à i-ésima raça.

É garantido que a soma de todos os e_i não ultrapasse 10^5.

Saída

Imprima uma única linha contendo o número de pares o qual Vasya consegue ligar dentro das possibilidades anotadas no canto da prova.

Exemplo de entrada	**Exemplo de saída**
3 3 1 1 2 1 3 0	2

F: Cardápio do RUFNl

`Arquivo: cardapio.[c|cpp|java|pas]`

A Universidade Federal da Nlogônia (a UFNl) dispõe de um Restaurante Universitário (o RUFNl) que fornece refeições diárias a todos os alunos da universidade.

O cardápio do restaurante é sempre composto por três alimentos distintos: uma entrada, um acompanhamento e uma sobremesa. Existem N alimentos que podem ser utilizados pelo restaurante para formar seu cardápio. Os alimentos são bem variados, de forma que qualquer alimento pode ser utilizado como entrada, como acompanhamento ou como sobremesa.

Entretanto, os nutricionistas exigem que o cardápio seja montado sempre de forma balanceada. Para tal, é necessário que a entrada tenha menos calorias que o acompanhamento e que o acompanhamento tenha menos calorias que a sobremesa. Além disso, é necessário que a entrada tenha mais fibras que o acompanhamento e que o acompanhamento tenha mais fibras que a sobremesa.

Sua tarefa é determinar quantos cardápios distintos podem ser formados, isto é, de quantas maneiras é possível selecionar três alimentos dentre os N disponíveis para formar um cardápio válido.

Entrada

A primeira linha contém o inteiro N ($3 \leq N \leq 10^6$). Considere que os alimentos são numerados de 1 a N, em ordem crescente de calorias. Desta forma, considere que o alimento i tem i Calorias.

A segunda linha contém N inteiros distintos $f_1, f_2, ..., f_N$ ($1 \leq f_i \leq N$), indicando a quantidade de fibra nos alimentos. O alimento i possui f_i unidades de fibra.

Saída

Imprima uma linha contendo o número de cardápios que podem ser formados.

Exemplo de entrada	**Exemplo de saída**
3 3 2 1	1

Exemplo de entrada	**Exemplo de saída**
4 3 4 2 1	2

Exemplo de entrada	**Exemplo de saída**
6 6 2 4 1 5 3	4

G: Balas de Morango

`Arquivo: balas.[c|cpp|java|pas]`

Os N netos da madame Beauvoir podem ser travessos, mas também podem ser educados às vezes. Um dia, eles fizeram uma grande bagunça com as portas de sua mansão! No dia seguinte, entretanto, eles se comportaram muito bem. Por isso, Madame Beauvoir decidiu recompensar seus netos com balas de morango.

Agora ela está na distribuidora de doces para comprar as balas. O número total de balas compradas deve ser tal que (i) todas as balas compradas devem ser distribuídas entre seus netos; e (ii) todos os netos devem receber a mesma quantidade de balas. Note que a quantidade de balas que cada neto recebe não é relevante, desde que todos recebam a mesma quantidade.

Há M pacotes de doces disponíveis na loja. O pacote i ($1 \leq i \leq M$) contém B_i balas de morango. O vendedor exige que Madame Beauvoir compre apenas pacotes inteiros, isto é, não é possível abrir um pacote na loja e comprar apenas algumas balas dentro dele.

Como ela é muito rica, ela não se importa com o preço de cada pacote. De fato, para impressionar seus netos, ela quer comprar o maior número possível de pacotes de balas. Sua tarefa é determinar qual é o maior número possível de pacotes que podem ser comprados por ela de forma que o total de balas compradas pode ser dividido entre seus netos.

Entrada

A primeira linha contém dois inteiros N e M ($1 \leq N, M \leq 10^3$). A segunda linha contém M inteiros B_i ($1 \leq B_i \leq 10^9$), indicando a quantidade de balas em cada pacote.

Saída

Imprima uma linha contendo o maior número possível de pacotes que podem ser comprados.

Exemplo de entrada	**Exemplo de saída**
`10 3` `3 4 7`	`2`

Exemplo de entrada	**Exemplo de saída**
`10 4` `12 7 2 6`	`3`

H: Robô Aspirador

`Arquivo: aspirador.[c|cpp|java|pas]`

Ricciardi, o robô aspirador, recebeu ordens. Deve limpar o máximo possível dos N grãos de sujeira no chão e chegar à estação de recarga. Parece uma tarefa trivial, mas Ricciardi está com a bateria viciada e pode realizar apenas M movimentos antes de esgotá-la.

Localizado em uma sala retangular dividida em $W \times H$ células quadradas, o robô pode se movimentar para as células adjacentes diretamente acima, abaixo, à esquerda e à direita de sua posição atual, desde que não haja obstáculos nela. Determinado a economizar energia e realizar seu trabalho com maestria, Ricciardi pediu a você para escrever um programa que calcule o número máximo de grãos de sujeira que Ricciardi consegue limpar.

Entrada

A primeira linha contém dois inteiros N ($1 \leq N \leq 8$) e M ($1 \leq M \leq 10^9$). A segunda linha também contém dois inteiros W e H ($5 \leq W, H \leq 100$).

As H linhas seguintes contém W caracteres cada e descrevem a sala. Obstáculos são representados por `#`, posições livres por `.`, a posição inicial de Ricciardi por `R`, grãos de sujeira por `*` e a estação de recarga por `S`.

Ricciardi coleta os grãos automaticamente ao passar por cima deles e consegue andar sobre a estação de recarga como em qualquer posição livre.

Saída

Imprima uma única linha com um único inteiro, o número máximo de grãos que Ricciardi consegue coletar chegando à estação de recarga. Se o robô não consegue chegar à estação, imprima `-1`.

Exemplo de entrada	**Exemplo de saída**

`3 10`
`5 5`
`S...*`
`.....`
`..*..`
`.....`
`*...R`
	`1`

Exemplo de entrada	Exemplo de saída
3 12 5 5 S...* ....*.. ..... *...R	2

Exemplo de entrada	Exemplo de saída
4 3 7 6 R*..... **..... ....S... ..*....	-1

I: Circo de Pulgas

`Arquivo: circo.[c|cpp|java|pas]`

Vasya e Petya resolveram investir em um novo empreendimento, abriram um circo de pulgas.

Neste exato momento estão ensaiando com suas P pulgas em uma linha reta. Cada inseto carrega uma pequena placa com um número inteiro e sabe seguir 4 ordens diferentes:

- Trocar o número em sua placa por outro especificado;
- Gritar a quantidade de números pares nas placas das pulgas de um dado intervalo de posições;
- Gritar a quantidade de números ímpares nas placas das pulgas de um dado intervalo de posições;
- Começar a dançar enquanto as pulgas de um dado intervalo de posições invertem a ordem de suas placas.

Alguns animais não entenderam a última tarefa e pediram um exemplo, Vasya prosseguiu:

Suponha que existam 10 pulgas inicialmente segurando placas com os seguintes valores:

```
70 15 3 4 15 59 0 1 444 2
```

Se a ordem for para inverter a ordem das placas no intervalo $[3, 7]$, a nova configuração seria:

```
70 15 0 59 15 4 3 1 444 2
```

A dupla pediu para que você escrevesse um programa que simule o ensaio das pulgas, desta forma eles podem verificar se elas estão bem treinadas.

Entrada

A primeira linha contém dois inteiros P e Q ($1 \leq P, Q \leq 10^5$), a linha seguinte contém P inteiros p_i ($0 \leq p_i \leq 10^9$) separados por espaços com o número inicialmente escrito na placa das pulgas.

As Q linhas seguintes descrevem as ordens dadas aos insetos e seguem uma das quatro possibilidades:

S a v Trocar o número da placa da pulga na posição a ($1 \leq a \leq P$) por v;

E a b Gritar a quantidade de números pares nas placas das pulgas no intervalo $[a, b]$ ($1 \leq a, b \leq P$);

O a b Gritar a quantidade de números ímpares nas placas das pulgas no intervalo $[a, b]$ ($1 \leq a \leq b \leq P$);

I a b Inverter a ordem das placas no intervalo $[a, b]$ ($1 \leq a \leq b \leq P$).

Saída

Para cada operação de grito (tanto `E` quanto `O`), imprima uma linha com o valor a ser gritado.

Exemplo de entrada	**Exemplo de saída**
7 8 30 50 1 2 3 4 10 O 1 10 S 2 7 E 1 10 S 3 8 O 1 10 O 1 3 I 3 5 O 1 3	2 7 2 1 2

J: Vírus Nlogono

`Arquivo: virus.[c|cpp|java|pas]`

A secretaria de saúde pública da Nlogônia acabou de emitir um alerta. Um vírus está contagiando toda a população.

Após muitos estudos, os pesquisadores do país determinaram que após infiltrarem o corpo hospedeiro, os virus se juntam dois a dois para tornarem-se letais.

O nível de letalidade de uma infecção é determinado pela soma da diferença da idade em dias dos vírus pareados. Os sem pares não influenciam no aumento do nível.

Desta forma, se existem 4 vírus no corpo hospedeiro com idades (em dias):

```
4 10 9 43
```

E eles se paream da seguinte forma:

```
4 9
43 10
```

O nível de letalidade seria 38 $((9-4)+(43-10))$.

A secretaria de saúde pública da Nlogônia pediu para que você escrevesse um programa que, dado a contagem de vírus em um corpo e a idade de cada um deles, calcule o nível máximo de letalide que a infecção pode assumir.

Entrada

A primeira linha contém um inteiro N $(1 \leq N \leq 1000)$, a quantidade de vírus no corpo hospedeiro. A linha seguinte contém N números inteiros a_i $(0 \leq a_i \leq 1000)$ separados por espaços, as idades (em dias) de todos os vírus no corpo.

Saída

Imprima uma única linha contendo o nível de letalidade máximo que a infecção pode assumir.

Exemplo de entrada	**Exemplo de saída**
4 4 9 43 10	40

Exemplo de entrada	**Exemplo de saída**
3 0 100 50	100