

ALGORITMOS I

7ª LISTA DE EXERCÍCIOS

1) QUERMESSE (OBI 2000)

Os alunos do último ano resolveram organizar uma quermesse para arrecadar fundos para a festa de formatura. A festa prometia ser um sucesso, pois o pai de um dos formandos, Teófilo, dono de uma loja de informática, decidiu doar um computador para ser sorteado entre os que comparecessem. Os alunos prepararam barracas de quentão, pipoca, doces, ensaiaram a quadrilha e colocaram à venda ingressos numerados sequencialmente a partir de 1. O número do ingresso serviria para o sorteio do computador. Ficou acertado que Teófilo decidiria o método de sorteio; em princípio o sorteio seria, claro, computadorizado.

O local escolhido para a festa foi o ginásio da escola. A entrada dos participantes foi pela porta principal, que possui uma roleta, onde passa uma pessoa por vez. Na entrada, um funcionário inseriu, em uma lista no computador da escola, o número do ingresso, na ordem de chegada dos participantes. Depois da entrada de todos os participantes, Teófilo começou a trabalhar no computador para preparar o sorteio. Verificando a lista de presentes, notou uma característica notável: havia apenas um caso, em toda a lista, em que o participante que possuía o ingresso numerado com i , havia sido a i -ésima pessoa a entrar no ginásio. Teófilo ficou tão encantado com a coincidência que decidiu que o sorteio não seria necessário: esta pessoa seria o ganhador do computador.

Tarefa

Conhecendo a lista de participantes, por ordem de chegada, sua tarefa é determinar o número do ingresso premiado, sabendo que o ganhador é o único participante que tem o número do ingresso igual à sua posição de entrada na festa.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém um número inteiro positivo N que indica o número de participantes da festa. A linha seguinte contém a sequência, em ordem de entrada, dos N ingressos das pessoas que participaram da festa. O final da entrada é indicado quando $N = 0$. Para cada conjunto de teste da entrada haverá um único ganhador.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas. A primeira linha identifica o conjunto de teste, no formato "Teste n ", onde n é numerado a partir de 1. A segunda linha deve conter o número do ingresso do ganhador, conforme determinado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigoro-

samente.

Exemplos

Exemplo de entrada 4 4 5 3 1 10 9 8 7 6 1 4 3 2 12 10 0	Exemplo de saída Teste 1 3 Teste 2 10
--	---

Restrições

$0 \leq N \leq 10000$ ($N = 0$ apenas para indicar o fim da entrada)

2) SALDO DE GOLS (OBI 2000)

Hipólito é um torcedor fanático. Coleciona flâmulas, bandeiras, recortes de jornal, figurinhas de jogadores, camisetas e tudo o mais que se refira a seu time preferido. Quando ganhou um computador de presente em uma festa, resolveu montar um banco de dados com os resultados de todos os jogos de seu time ocorridos desde a sua fundação, em 1911. Depois de inseridos os dados, Hipólito começou a ficar curioso sobre estatísticas de desempenho do time. Por exemplo, ele deseja saber qual foi o período em que o seu time acumulou o maior saldo de gols. Como Hipólito tem o computador há muito pouco tempo, não sabe programar muito bem, e precisa de sua ajuda.

Tarefa

É dada uma lista, numerada seqüencialmente a partir de 1, com os resultados de todos os jogos do time (primeira partida: 3 x 0, segunda partida: 1 x 2, terceira partida: 0 x 5 ...). Sua tarefa é escrever um programa que determine em qual período o time conseguiu acumular o maior saldo de gols. Um *período* é definido pelos números de seqüência de duas partidas, A e B , onde $A \leq B$. O saldo de gols acumulado entre A e B é dado pela soma dos gols marcados pelo time em todas as partidas realizadas entre A e B (incluindo as mesmas) menos a soma dos gols marcados pelos times adversários no período. Se houver mais de um período com o mesmo saldo de gols, escolha o maior período (ou seja, o período em que $B - A$ é maior). Se ainda assim houver mais de uma solução possível, escolha qualquer uma delas como resposta.

Entrada

Seu programa deve ler vários conjuntos de teste. A primeira linha de um conjunto de teste contém um inteiro não negativo, N , que indica o número de partidas realizadas pelo time (o valor $N = 0$ indica o final da entrada). Seguem-se N linhas, cada uma contendo um par de números inteiros não negativos X e Y que representam o resultado da partida: X são os gols a favor e Y os gols contra o time de Hipólito. As partidas são numeradas sequencialmente a partir de 1, na ordem em que aparecem na entrada.

Exemplo de Entrada

```
2
2 3
7 1
9
2 2
0 5
6 2
1 4
0 0
5 1
1 5
6 2
9
0 5
3
0 2
0 3
0 4
0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n ”, onde n é numerado a partir de 1. A segunda linha deve conter um par de inteiros I e J que indicam respectivamente a primeira e última partidas do melhor período, conforme determinado pelo seu programa, exceto quando o saldo de gols do melhor período for menor ou igual a zero; neste caso a segunda linha deve conter a expressão “nenhum”. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
2 2

Teste 2
3 8
```

Teste 3
nenhum

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq N \leq 10000$ ($N = 0$ apenas para indicar o fim da entrada)

$1 \leq A \leq N$

$A \leq B \leq N$

$0 \leq X \leq 50$

$0 \leq Y \leq 50$

3) BITS TROCADOS (OBI 2000)

As Ilhas Weblands formam um reino independente nos mares do Pacífico. Como é um reino recente, a sociedade é muito influenciada pela informática. A moeda oficial é o Bit; existem notas de B\$ 50,00, B\$10,00, B\$5,00 e B\$1,00. Você foi contratado(a) para ajudar na programação dos caixas automáticos de um grande banco das Ilhas Weblands.

Tarefa

Os caixas eletrônicos das Ilhas Weblands operam com todos os tipos de notas disponíveis, mantendo um estoque de cédulas para cada valor (B\$ 50,00, B\$10,00, B\$5,00 e B\$1,00). Os clientes do banco utilizam os caixas eletrônicos para efetuar retiradas de um certo número inteiro de Bits. Sua tarefa é escrever um programa que, dado o valor de Bits desejado pelo cliente, determine o número de cada uma das notas necessário para totalizar esse valor, de modo a minimizar a quantidade de cédulas entregues. Por exemplo, se o cliente deseja retirar B\$50,00, basta entregar uma única nota de cinquenta Bits. Se o cliente deseja retirar B\$72,00, é necessário entregar uma nota de B\$50,00, duas de B\$10,00 e duas de B\$1,00.

Entrada

A entrada é composta de vários conjuntos de teste. Cada conjunto de teste é composto por uma única linha, que contém um número inteiro positivo V , que indica o valor solicitado pelo cliente. O final da entrada é indicado por $V = 0$.

Exemplo de Entrada

1
72
0

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado a partir de 1. Na segunda linha devem aparecer quatro inteiros I, J, K e L que representam o resultado encontrado pelo seu programa: I indica o número de cédulas de B\$50,00, J indica o número de cédulas de B\$10,00, K indica o número de cédulas de B\$5,00 e L indica o número de cédulas de B\$1,00. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

Teste 1
0 0 0 1

Teste 2
1 2 0 2

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq V \leq 10000$ (V= 0 apenas para indicar o fim da entrada)

4) METEOROS (OBI 2001)

Em noites sem nuvens pode-se muitas vezes observar pontos brilhantes no céu que se deslocam com grande velocidade, e em poucos segundos desaparecem de vista: são as chamadas estrelas cadentes, ou *meteoros*. Meteoros são na verdade partículas de poeira de pequenas dimensões que, ao penetrar na atmosfera terrestre, queimam-se rapidamente (normalmente a uma altura entre 60 e 120 quilômetros). Se os meteoros são suficientemente grandes, podem não queimar-se completamente na atmosfera e dessa forma atingem a superfície terrestre: nesse caso são chamados de *meteoritos*.

Zé Felício é um fazendeiro que adora astronomia e descobriu um portal na Internet que fornece uma lista das posições onde caíram meteoritos. Com base nessa lista, e conhecendo a localização de sua fazenda, Zé Felício deseja saber quantos meteoritos caíram dentro de sua propriedade. Ele precisa de sua ajuda para escrever um programa de computador que faça essa verificação automaticamente.

Tarefa

São dados:

- uma lista de pontos no plano cartesiano, onde cada ponto corresponde à posição onde caiu um meteorito;
- as coordenadas de um retângulo que delimita uma fazenda.

As linhas que delimitam a fazenda são paralelas aos eixos cartesianos. Sua tarefa é escrever um programa que determine quantos meteoritos caíram dentro da fazenda (incluindo meteoritos que caíram exatamente sobre as linhas que delimitam a fazenda).

Entrada

Seu programa deve ler vários conjuntos de testes. A primeira linha de um conjunto de testes quatro números inteiros X_1 , Y_1 , X_2 e Y_2 , onde (X_1, Y_1) é a coordenada do canto superior esquerdo e (X_2, Y_2) é a coordenada do canto inferior direito do retângulo que delimita a fazenda. A segunda linha contém um inteiro, N , que indica o número de meteoritos. Seguem-se N linhas, cada uma contendo dois números inteiros X e Y , correspondendo às coordenadas de cada meteorito. O final da entrada é indicado por $X_1 = Y_1 = X_2 = Y_2 = 0$.

Exemplo de Entrada

```
2 4 5 1
2
1 2
3 3
2 4 3 2
3
1 1
2 2
3 3
0 0 0 0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n ”, onde n é numerado a partir de 1. A segunda linha deve conter o número de meteoritos que caíram dentro da fazenda. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
1

Teste 2
2
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$$0 \leq N \leq 10.000$$

$$0 \leq X \leq 10.000$$

$$0 \leq Y \leq 10.000$$

$$0 \leq X1 < X2 \leq 10.000$$

$$0 \leq Y2 < Y1 \leq 10.000$$

5) CALCULANDO (OBI 2001)

A disseminação dos computadores se deve principalmente à capacidade de eles se comportarem como outras máquinas, vindo a substituir muitas destas. Esta flexibilidade é possível porque podemos alterar a funcionalidade de um computador, de modo que ele opere da forma que desejarmos: essa é a base do que chamamos programação.

Tarefa

Sua tarefa é escrever um programa que faça com que o computador opere como uma calculadora simples. O seu programa deve ler expressões aritméticas e produzir como saída o valor dessas expressões, como uma calculadora faria. O programa deve implementar apenas um subconjunto reduzido das operações disponíveis em uma calculadora: somas e subtrações.

Entrada

A entrada é composta de vários conjuntos de testes. A primeira linha de um conjunto de testes contém um número inteiro m ($1 \leq m \leq 100$), indicando o número de operandos da expressão a ser avaliada. A segunda linha de um conjunto de testes contém a expressão aritmética a ser avaliada, no seguinte formato:

$X1 \ s1 \ X2 \ s2 \ \dots \ X_{m-1} \ s_{m-1} \ X_m$

Onde

- X_i , $1 \leq i \leq m$, é um operando ($0 \leq X_i \leq 100$);
- s_j , $1 \leq j < m$, é um operador, representado pelos símbolos '+' ou '-';
- não há espaços em branco entre operandos e operadores.

O final da entrada é indicado pelo valor $m=0$.

Saída

Para cada conjunto de testes da entrada seu programa deve produzir três linhas. A primeira linha deve conter um identificador da expressão, no formato "Teste n", onde

n é numerado a partir de 1. Na segunda linha deve aparecer o resultado encontrado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplos

Exemplo de entrada	Exemplo de saída
3	Teste 1
3+7-22	-12
3	
5-10-77	Teste 2
10	-82
1+2+3+4+5+6+7+8+9+10	
0	Teste 3
	55

Restrições

$$1 \leq m \leq 100$$

$$0 \leq X_i \leq 100 \text{ para todo } 1 \leq i \leq m$$

6) COFRINHOS DA VOVÓ (OBI 2003)

Vó Vitória mantém, desde o nascimento dos netos Joãozinho e Zezinho, um ritual que faz a alegria dos meninos. Ela guarda todas as moedas recebidas como troco em dois pequenos cofrinhos, um para cada neto. Quando um dos cofrinhos fica cheio, ela chama os dois netos para um alegre almoço, ao final do qual entrega aos garotos as moedas guardadas nos cofrinhos de cada um.

Ela sempre foi muito zelosa quanto à distribuição igualitária do troco arrecadado. Se, por força do valor das moedas, ela não consegue depositar a mesma quantia nos dois cofrinhos, ela memoriza a diferença de forma a compensá-la no próximo depósito.

Tarefa

Vó Vitória está ficando velha e tem medo que deslizos de memória a façam cometer injustiças com os netos, deixando de compensar as diferenças entre os cofrinhos. Sua tarefa é ajudar Vó Vitória, escrevendo um programa de computador que indique as diferenças entre os depósitos, de forma que ela não tenha que preocupar-se em memorizá-las.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém um número inteiro N, que indica o número de depósitos nos cofrinhos. As N linhas seguintes descrevem cada uma um depósito nos cofrinhos; o de-

pósito é indicado por dois valores inteiros J e Z, separados por um espaço em branco, representando respectivamente os valores, em centavos, depositados nos cofres de João e Zezinho. O final da entrada é indicado por N = 0.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir um conjunto de linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado seqüencialmente a partir de 1. A seguir seu programa deve escrever uma linha para cada depósito do conjunto de testes. Cada linha deve conter o um inteiro que representa a diferença (em centavos) entre o valor depositado no cofrinho do João e do Zezinho. Deixe uma linha em branco ao final de cada conjunto de teste. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplos

Exemplo de entrada	Exemplo de saída
3	Teste 1
20 25	-5
10 5	0
10 10	0
4	Teste 2
0 5	-5
10 0	5
0 10	-5
10 0	5
0	

Restrições

$0 \leq N \leq 100$ (N = 0 apenas para indicar o fim da entrada)
 $0 \leq J \leq 100$ (valor de cada depósito no cofre de Joãozinho)
 $0 \leq Z \leq 100$ (valor de cada depósito no cofre de Zezinho)

7) PAR OU ÍMPAR (OBI 2004)

Muitas crianças gostam de decidir todas as disputas através do famoso jogo de Par ou Ímpar. Nesse jogo, um dos participantes escolhe Par e o outro Ímpar. Após a escolha, os dois jogadores mostram, simultaneamente, uma certa quantidade de dedos de uma das mãos. Se a soma dos dedos das mãos dos dois jogadores for par, vence o jogador que escolheu Par inicialmente, caso contrário vence o que escolheu Ímpar.

Tarefa

Dada uma sequência de informações sobre partidas de Par ou Ímpar (nomes dos jogadores e números que os jogadores escolheram), você deve escrever um programa para indicar o vencedor de cada uma das partidas.

Entrada

A entrada é composta de vários conjuntos de testes. A primeira linha de um conjunto de testes contém um inteiro N , que indica o número de partidas de Par ou Ímpar que aconteceram. As duas linhas seguintes contêm cada uma um nome de jogador. Um nome de jogador é uma cadeia de no mínimo um e no máximo dez letras (maiúsculas e minúsculas), sem espaços em branco. As N linhas seguintes contêm cada uma dois inteiros A e B que representam o número de dedos que cada jogador mostrou em cada partida ($0 \leq A \leq 5$ e $0 \leq B \leq 5$). Em todas as partidas, o primeiro jogador sempre escolhe Par. O final da entrada é indicado por $N = 0$.

Exemplo de Entrada

```
3
Pedro
Paulo
2 4
3 5
1 0
2
Claudio
Carlos
1 5
2 3
0
```

Saída

Para cada conjunto de teste da entrada, seu programa deve produzir a saída da seguinte forma. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n ”, onde n é numerado seqüencialmente a partir de 1. As próximas N linhas devem indicar o nome do vencedor de cada partida. A próxima linha deve ser deixada em branco. A grafia mostrada no Exemplo de 3 Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
Pedro
Pedro
Paulo

Teste 2
Claudio
Carlos
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq N \leq 1000$ ($N = 0$ apenas para indicar o fim da entrada)

$0 \leq A \leq 5$

$0 \leq B \leq 5$

$1 \leq \text{comprimento do nome de jogador} \leq 10$

8) FROTA DE TÁXI (OBI 2005)

A Companhia de Táxi Tabajara (CTT) é uma das maiores empresas de transporte do país. Possui uma vasta frota de carros e opera em todas as grandes cidades. Recentemente a CTT modernizou a sua frota, adquirindo um lote de 500 carros bi-combustíveis (carros que podem utilizar como combustível tanto álcool quanto gasolina). Além do maior conforto para os passageiros e o menor gasto com manutenção, com os novos carros é possível uma redução adicional de custo: como o preço da gasolina está sujeito a variações muito bruscas e pode ser vantagem, em certos momentos, utilizar álcool como combustível. Entretanto, os carros possuem um melhor desempenho utilizando gasolina, ou seja, em geral, um carro percorre mais quilômetros por litro de gasolina do que por litro de álcool.

Tarefa

Você deve escrever um programa que, dados o preço do litro de álcool, o preço do litro de gasolina e os quilômetros por litro que um carro bi-combustível realiza com cada um desses combustíveis, determine se é mais econômico abastecer os carros da CTT com álcool ou com gasolina. No caso de não haver diferença de custo entre abastecer com álcool ou gasolina a CTT prefere utilizar gasolina.

Entrada

A entrada é composta por uma linha contendo quatro números reais com precisão de duas casas decimais A , G , R_a e R_g , representando respectivamente o preço por litro do álcool, o preço por litro da gasolina, o rendimento (km/l) do carro utilizando álcool e o rendimento (km/l) do carro utilizando gasolina.

A entrada deve ser lida do dispositivo de entrada padrão (normalmente o teclado).

Saída

A saída deve ser composta por uma única linha contendo o caractere 'A' se é mais econômico abastecer a frota com álcool ou o caractere 'G' se é mais econômico ou indiferente abastecer a frota com gasolina.

A saída deve ser escrita no dispositivo de saída padrão (normalmente a tela).

Restrições

$$0.01 \leq A \leq 10.00$$

$$0.01 \leq G \leq 10.00$$

$$0.01 \leq R_a \leq 20.00$$

$$0.01 \leq R_g \leq 20.00$$

9) BAFO (OBI 2005)

Álbuns de figurinhas – sejam de times de futebol, princesas ou super-heróis – tem marcado gerações de crianças e adolescentes. Conseguir completar um álbum é uma tarefa muitas vezes árdua, envolvendo negociações com colegas para a troca de figurinhas. Mas a existência das figurinhas propicia uma outra brincadeira, que foi muito popular entre crianças no século passado: o jogo de *bater figurinhas* (o famoso “Bafo”). O jogo é muito simples, mas divertido (e muito competitivo). No início de uma partida, cada criança coloca em uma pilha um certo número de figurinhas. Uma partida é composta de rodadas; a cada rodada as crianças batem com a mão sobre a pilha de figurinhas, tentando virá-las com o vácuo formado pelo movimento da mão. As crianças jogam em turnos, até que a pilha de figurinhas esteja vazia. Ganha a partida a criança que conseguir virar mais figurinhas.

Aldo e Beto estão jogando bafo com todas as suas figurinhas e pediram sua ajuda para calcular quem é o vencedor.

Tarefa

Você deve escrever um programa que, dada a quantidade de figurinhas que Aldo e Beto viraram em cada rodada, determine qual dos dois é o vencedor.

Entrada

A entrada é composta de vários casos de teste, cada um correspondendo a uma partida entre Aldo e Beto. A primeira linha de um caso de teste contém um número inteiro R que indica quantas rodadas ocorreram na partida. Cada uma das R linhas seguintes contém dois inteiros, A e B , que correspondem, respectivamente, ao número de figurinhas que Aldo e Beto conseguiram virar naquela rodada. Em todos os casos de teste há um único vencedor (ou seja, não ocorre empate). O final da entrada é indicado por $R = 0$.

A entrada deve ser lida do dispositivo de entrada padrão (normalmente o teclado).

Saída

Para cada caso de teste da entrada, seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do caso de teste, no formato “Teste n ”, onde n é numerado sequencialmente a partir de 1. A segunda linha deve conter

o nome do vencedor (Aldo ou Beto). A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

A saída deve ser escrita no dispositivo de saída padrão (normalmente a tela).

Restrições

$1 \leq R \leq 1000$ ($R = 0$ apenas para indicar o final da entrada)

$0 \leq A \leq 100$

$0 \leq B \leq 100$

Exemplo de Entrada

```
2
1 5
2 3
3
0 0
4 7
10 0
0
```

Saída para o Exemplo de Entrada

```
Teste 1
Beto
```

```
Teste 2
Aldo
```

10) CONVERSA NÃO TÃO SECRETA (OBI 2006)

A polícia desconfia que dois homens que passeiam todos os dias pelo parque são na verdade criminosos. O parque é plano, de formato retangular, e estreitas faixas de grama o dividem em quadrados de mesmo tamanho, formando uma grade de N por M quadrados.

Os dois homens tem um comportamento curioso e suspeito em seu passeio: após encontrarem-se, conversam durante um minuto, andam mudando rapidamente de lugar, passando a ocupar um novo quadrado do parque, conversam mais um minuto, andam novamente (mudando de quadrado), conversam mais um minuto, e assim sucessivamente. A cada minuto escolhem uma direção (Norte, Sul, Leste ou Oeste) e andam até o quadrado imediatamente vizinho na direção escolhida.

Tentando escutar trechos das conversas dos homens, a polícia instalou um pequeno microfone multi-direcional em um dos quadrados do parque. O microfone é capaz de captar conversas realizadas no quadrado onde está instalado e em todos os quadrados imediatamente vizinhos.

Os dois homens sempre iniciam o passeio no quadrado de coordenadas (0,0).

Tarefa

Dadas as coordenadas do microfone e a sequência de movimentos que os dois homens realizaram durante seu passeio no parque, seu programa deve determinar quantos minutos de conversa foram captados pelo microfone.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). A primeira linha contém dois inteiros N e M que indicam respectivamente o número de linhas e o número de colunas do parque ($0 \leq N \leq 1000000$ e $0 \leq M \leq 1000000$). A segunda linha contém dois inteiros X e Y que indicam a coordenada do microfone em termos de linhas e colunas ($0 \leq X \leq N$ e $0 \leq Y \leq M$). A terceira linha contém um inteiro K , indicando o número de quadrados pelos quais os dois homens passearam. A quarta linha contém K inteiros, entre 1, 2, 3, 4, que indicam a rota tomada pelos dois homens durante o passeio; cada inteiro indica a direção tomada ao final de um minuto de conversa, com 1 representando o Norte, 2 representando o Sul, 3 representando o Leste e 4 representando o Oeste.

Saída

Seu programa deve imprimir, na saída padrão, uma única linha contendo um inteiro: o número de minutos de conversação captados pelo microfone.

Exemplos

Entrada 10 10 2 2 3 3 3 3 Saída 0	Entrada 5 5 0 1 3 3 1 3 Saída 3	20 20 3 2 8 1 1 3 3 1 1 2 4 Saída 6
---	---	--

11) PÃO A METRO (OBI 2007)

Pão a metro é um tipo de sanduíche gigante que é uma excelente opção de lanche para torneios de programação, embora a experiência já tenha mostrado que o oferecimento de sanduíches pode gerar reclamação dos competidores. Outro grande problema é que algumas pessoas são mais gulosas que outras e, dessa maneira,

acabam pegando pedaços maiores que os pedaços dos outros. Para a final da OBI, a coordenação estava pensando em providenciar pão a metro para os competidores, porém tais problemas os fizeram recuar na idéia.

Embora a idéia tenha sido momentaneamente abandonada, uma idéia simples surgiu: cortar previamente o pão em fatias de tamanho iguais e distribuí-las entre as pessoas. O único problema com tal idéia é que se o número de pessoas for muito grande, fica impraticável ter apenas um pão. Por exemplo, se quisermos que 1.000 pessoas recebam 20 centímetros de sanduíche, seria necessário um sanduíche de 20.000 centímetros, ou 200 metros!

Alguém levantou a seguinte hipótese: se houvessem N pessoas e fossem encomendados K sanduíches de empresas diferentes, cada qual com uma determinada metragem (tamanho) M_i ($1 \leq i \leq K$), seria possível retirar desses pães N fatias de mesmo tamanho, possivelmente sobrando partes não utilizadas. A questão seria: qual o tamanho inteiro máximo que essas fatias poderão ter?

Por exemplo, se tivermos $K = 4$, com os tamanhos (em centímetros) $M_1 = 120$, $M_2 = 89$, $M_3 = 230$ e $M_4 = 177$ e $N = 10$, podemos retirar N fatias iguais de tamanho máximo 57, pois assim conseguimos 2 fatias no primeiro pão, 1 no segundo, 4 no terceiro e 3 no quarto, totalizando as 10 fatias necessárias. Se tentarmos cortar fatias de tamanho 58, só será possível obter 3 fatias do terceiro pão, totalizando 9 e, portanto, 57 é realmente o melhor que podemos obter. Note que não podemos usar duas ou mais fatias menores de diferentes pães para formarmos uma fatia do tamanho selecionado. (ficaria muito deselegante dar um lanche recortado às pessoas).

Tarefa

Escreva um programa que, dados os tamanhos de pão disponíveis (em centímetros) e a quantidade de pessoas a serem atendidas, retorne o tamanho inteiro máximo (em centímetros) da fatia que pode ser cortada de maneira a atender todas as pessoas.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). A primeira linha da entrada contém um inteiro N que indica a quantidade pessoas ($1 \leq N \leq 10.000$). A segunda linha contém um inteiro K ($1 \leq K \leq 10.000$) que é a quantidade de sanduíches disponível. Na terceira linha há K inteiros M ($1 \leq M \leq 10.000$) separados por um espaço em branco representando o tamanho de cada pão.

Saída

Seu programa deve imprimir, na saída padrão, uma única linha, contendo o tamanho inteiro máximo da fatia que pode ser cortada.

Exemplos

Exemplo de entrada 10 4 120 89 230 177	Exemplo de saída 57
Exemplo de entrada 3 2 45 85	Exemplo de saída 42
Exemplo de entrada 7 7 100 98 99 505 102 97 101	Exemplo de saída 101

12) OBI

O principal prêmio da Olimpíada Brasileira de Informática é o convite para os cursos de programação oferecidos no Instituto de Computação da Unicamp, com todas as despesas pagas pela Fundação Carlos Chagas, patrocinadora da OBI. São convidados apenas os competidores que atingem um certo número mínimo de pontos, consideradas as duas fases de provas.

Você foi contratado pela Coordenação da OBI para fazer um programa que, dados os números de pontos obtidos por cada competidor em cada uma das fases, e o número mínimo de pontos para ser convidado, determine quantos competidores serão convidados para o curso na Unicamp.

Você deve considerar que:

- todos os competidores participaram das duas fases;
- o total de pontos de um competidor é a soma dos pontos obtidos nas duas fases;

Por exemplo, se a pontuação mínima para ser convidado é 435 pontos, um competidor que tenha obtido 200 pontos na primeira fase e 235 pontos na segunda fase será convidado para o curso na Unicamp. Já um competidor que tenha obtido 200 pontos na primeira fase e 234 pontos na segunda fase não será convidado.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). A primeira linha da entrada contém dois

números inteiros N e P , representando respectivamente o número de competidores ($1 \leq N \leq 1000$) e o número mínimo de pontos para ser convidado ($1 \leq P \leq 1000$). Cada uma das N linhas seguintes contém dois números inteiros X e Y indicando a pontuação de um competidor em cada uma das fases ($0 \leq X \leq 400$) e ($0 \leq Y \leq 400$).

Saída

Seu programa deve imprimir na saída padrão uma única linha contendo um único inteiro, indicando o número de competidores que serão convidados a participar do curso na Unicamp.

Exemplos

<p>Exemplo de entrada</p> <pre>3 100 50 50 100 0 49 50</pre>	<p>Exemplo de saída</p> <pre>2</pre>
<p>Exemplo de entrada</p> <pre>4 235 100 134 0 0 200 200 150 150</pre>	<p>Exemplo de saída</p> <pre>2</pre>

13) SEDEX MARCIANO (OBI 2010)

Estamos no ano 2048 e um dos sonhos da humanidade torna-se finalmente realidade: a colonização do planeta Marte. Nossos primeiros colonizadores acabam de chegar, e já começam a fazer as preparações (como a instalação de cúpulas de oxigênio e tratamento do solo para agricultura) para que mais pessoas possam tentar uma nova vida no planeta vizinho.

Apesar dos avanços tecnológicos e desafios vencidos, ainda resta um grande problema: os foguetes usados para ir a Marte ainda são complicados e caros. Com isso, fica difícil enviar suprimentos para os nossos colonos (enquanto a agricultura ainda não é possível) por muito tempo. Assim, a agência espacial contratou o SBC (Serviço Balístico Cósmico), que desenvolveu um canhão super-potente que consegue disparar esferas até Marte, sem precisar gastar milhões de dólares em equipamento e combustível.

Agora, tudo o que é necessário fazer para enviar suprimentos a Marte é colocar uma caixa com as encomendas dentro de uma esfera e disparar a mesma até seu destino.

Tarefa

Dadas as dimensões de uma caixa com suprimentos e o raio interno da esfera que é disparada pelo canhão, seu programa deverá dizer se é possível enviar tal caixa para Marte usando tal esfera.

Entrada

Cada entrada contém apenas uma linha com quatro inteiros L, A, P e R, ($0 \leq L, A, P, R \leq 1000$) que representam, respectivamente, a largura, altura e profundidade da caixa, e o raio da esfera.

Saída

Seu programa deve imprimir um único caractere: 'S' (sem aspas) se é possível colocar a caixa dentro da esfera, ou 'N' (sem aspas) caso contrário.

Exemplos

Exemplo de entrada 10 20 30 30	Exemplo de saída S
Exemplo de entrada 10 10 10 7	Exemplo de saída N
Exemplo de entrada 2 4 4 3	Exemplo de saída S

14) O MAR NÃO ESTÁ PARA PEIXE (OBI 2011)

Em um arquipélago no meio do Oceano Pacífico a economia é regida pela pesca, pois o peixe é o principal alimento disponível. Ultimamente, a população desse arquipélago tem aumentado drasticamente, o que levou a um grande aumento da pesca, e, conseqüentemente, a problemas.

Neste arquipélago, cada pescador vai diariamente ao alto mar com a intenção de conseguir trazer o maior número de peixes para o seu vilarejo. Com a expansão da pesca, os pescadores estão começando a jogar suas redes de pesca por cima das de outros pescadores. Com isso, os pescadores perdem, pois apenas o primeiro pescador pega os peixes da intersecção entre as redes.

A Associação dos Pescadores da ilha decidiu fazer um levantamento para descobrir

quanto do mar está de fato sendo aproveitado, ou seja, qual a área do mar que está coberta por pelo menos uma rede de pesca.

Como há muitas intersecções entre as redes de pesca, é muito difícil para a associação calcular a área total da região coberta pelas redes. Por este motivo, eles pediram para que você escrevesse um programa para resolver este problema.

Como é muito difícil navegar pelo mar, os pescadores sempre jogam as redes de forma que as regiões cobertas por cada rede são sempre retângulos com lados paralelos aos eixos, se imaginarmos o mar como um plano cartesiano.

Entrada

A primeira linha da entrada possui um inteiro N indicando o número de redes que foram lançadas. As próximas N linhas descrevem as regiões cobertas pelas redes: cada uma contém quatro inteiros X_i e X_f , Y_i e Y_f . A região coberta pela rede em questão contém todo ponto (X, Y) tal que $X_i \leq X \leq X_f$ e $Y_i \leq Y \leq Y_f$.

Saída

A saída deve conter apenas uma linha contendo a área da região do mar realmente aproveitada pelos pescadores, ou seja, a área total da região do mar coberta por pelo menos uma rede de pesca.

Restrições

- $1 \leq N \leq 100$
- $1 \leq X_i \leq X_f \leq 100$
- $1 \leq Y_i \leq Y_f \leq 100$

Exemplos

Entrada 2 1 3 1 3 2 4 2 4	Saída 7
Entrada 3 1 6 1 2 3 7 1 2 2 5 1 2	Saída 6

15) COLCHÃO (OBI 2012)

João está comprando móveis novos para sua casa. Agora é a vez de comprar um colchão novo, de molas, para substituir o colchão velho. As portas de sua casa têm altura H e largura L e existe um colchão que está em promoção com dimensões $A \times B \times C$.

O colchão tem a forma de um paralelepípedo reto retângulo e João só consegue arrastá-lo através de uma porta com uma de suas faces paralelas ao chão, mas consegue virar e rotacionar o colchão antes de passar pela porta.

Entretanto, de nada adianta ele comprar o colchão se ele não passar através das portas de sua casa. Portanto ele quer saber se consegue passar o colchão pelas portas e para isso precisa de sua ajuda.

Entrada

A primeira linha da entrada contém três números inteiros A , B e C , as três dimensões do colchão, em centímetros. A segunda linha contém dois inteiros H e L , respectivamente a altura e a largura das portas em centímetros.

Saída

Seu programa deve escrever uma única linha, contendo apenas a letra 'S' se o colchão passa pelas portas e apenas a letra 'N' em caso contrário.

Restrições

- $1 \leq A, B, C \leq 300$
- $1 \leq H, L \leq 250$

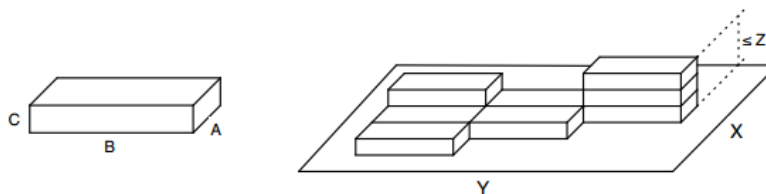
Exemplos

Exemplo de entrada 25 120 220 200 100	Exemplo de saída S
Exemplo de entrada 25 205 220 200 100	Exemplo de saída N
Exemplo de entrada 25 200 220 200 100	Exemplo de saída S

16) TRANSPORTE DE CONTÊINERES (OBI 2011)

A Betalândia é um país que apenas recentemente se abriu para o comércio exterior e está preparando agora sua primeira grande exportação. A Sociedade Betalandesa de Comércio (SBC) ficou encarregada de conduzir a exportação e determinou que, seguindo os padrões internacionais, a carga será transportada em contêineres, que são, por sua vez, colocados em grandes navios para o transporte internacional.

Todos os contêineres betalandeses são idênticos, medindo A metros de largura, B metros de comprimento e C metros de altura. Um navio porta-contêineres pode ser visto como um retângulo horizontal de X metros de largura e Y metros de comprimento, sobre o qual os contêineres são colocados. Nenhuma parte de contêiner pode ficar para fora do navio. Além disso, para possibilitar a travessia de pontes, a altura máxima da carga no navio não pode ultrapassar Z metros.



Devido a limitações do guindaste utilizado, os contêineres só podem ser carregados alinhados com o navio. Ou seja, os contêineres só podem ser colocados sobre o navio de tal forma que a largura e o comprimento do contêiner estejam paralelos à largura e ao comprimento do navio, respectivamente.

A SBC está com problemas para saber qual a quantidade máxima de contêineres que podem ser colocados no navio e pede sua ajuda. Sua tarefa, neste problema, é determinar quantos contêineres podem ser carregados no navio respeitando as restrições acima.

Entrada

A entrada consiste de duas linhas. A primeira linha contém três inteiros A , B e C que representam as dimensões dos contêineres, enquanto a segunda linha contém outros três inteiros X , Y e Z que representam as dimensões do navio.

Saída

Seu programa deve imprimir apenas uma linha contendo um inteiro que indica a quantidade máxima de contêineres que o navio consegue transportar.

Restrições

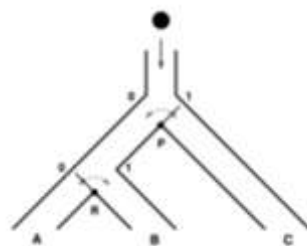
- $1 \leq A, B, C, X, Y, Z \leq 10^6$
- É garantido que a maior resposta será menor ou igual a 10^6 .

Exemplos

Entrada 1 1 1 1 1 1	Saída 1
Entrada 1 2 5 9 6 11	Saída 54
Entrada 1 2 12 6 9 10	Saída 0

17) FLÍPER (OBI 2015)

Flíper é um tipo de jogo onde uma bolinha de metal cai por um labirinto de caminhos até chegar na parte de baixo do labirinto. A quantidade de pontos que o jogador ganha depende do caminho que a bolinha seguir. O jogador pode controlar o percurso da bolinha mudando a posição de algumas portinhas do labirinto. Cada portinha pode estar na posição 0, que significa virada para a esquerda, ou na posição 1 que quer dizer virada para a direita. Considere o flíper da figura abaixo, que tem duas portinhas. A portinha P está na posição 1 e a portinha R, na posição 0. Desse jeito, a bolinha vai cair pelo caminho B.



Você deve escrever um programa que, dadas as posições das portinhas P e R, neste flíper da figura, diga por qual dos três caminhos, A, B ou C, a bolinha vai cair!

Entrada

A entrada é composta por apenas uma linha contendo dois números P e R, indicando as posições das duas portinhas do flíper da figura.

Saída

A saída do seu programa deve ser também apenas uma linha, contendo uma letra maiúscula que indica o caminho por onde a bolinha vai cair: 'A', 'B' ou 'C'.

Restrições

- O número P pode ser 0 ou 1. O número R pode ser 0 ou 1.

Exemplos

Entrada 1 0	Saída B
Entrada 0 0	Saída C

18) MATRIZ ESCADA (OBI 2015)

Joãozinho está aprendendo sobre matrizes. Hoje ele aprendeu como deixar matrizes na forma escada, e está exercitando. Para ajudá-lo, você deve escrever um programa que determine se o resultado dele realmente está no formato correto.

Uma matriz está na forma escada quando, para cada linha, as condições a seguir forem satisfeitas:

- Se a linha só possuir zeros, então todas as linhas abaixo desta também só possuem zeros.
- Caso contrário, seja X o elemento diferente de zero mais à esquerda da linha; então, para todas as linhas abaixo da linha de X, todos os elementos nas colunas à esquerda de X e na coluna de X são iguais a zero.

Entrada

A primeira linha possui dois inteiros N e M, as dimensões da matriz. Cada uma das N linhas seguintes contém M inteiros não-negativos, os elementos da matriz.

Saída

Seu programa deve produzir uma única linha, contendo o caractere 'S' caso a matriz esteja no formato escada, ou 'N', caso contrário.

Restrições

- $1 \leq N \leq 500$ e $1 \leq M \leq 500$.
- Cada elemento da matriz está entre 0 e 10^5 .

Exemplos

<p>Entrada</p> <p>4 6</p> <p>1 2 9 9 9 9</p> <p>0 0 3 9 9 9</p> <p>0 0 0 0 5 9</p> <p>0 0 0 0 0 6</p>	<p>Saída</p> <p>S</p>
<p>Entrada</p> <p>5 8</p> <p>0 5 1 0 3 2 2 0</p> <p>0 0 0 0 4 0 1 2</p> <p>0 0 0 0 0 0 3 2</p> <p>0 0 0 0 0 0 0 0</p> <p>0 0 0 0 0 0 0 0</p>	<p>Saída</p> <p>S</p>
<p>Entrada</p> <p>5 5</p> <p>1 1 2 3 4</p> <p>0 1 1 4 5</p> <p>0 1 2 3 6</p> <p>0 0 0 2 0</p> <p>0 0 0 0 0</p>	<p>Exemplo de saída</p> <p>N</p>