

15^o Treino para alunos da UFPR

22 de Maio de 2015



Flávio Zavan
Ricardo Oliveira

A: Cheddar

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

Pobre Petya está com fome.

– Mãe! Mãe! Mããããe! – Parte o grito do porão.

– Que foi, piá? – A doce senhora responde.

– Quero cheddar!

CHEDDAR! O menino está com fome, quer cheddar. Você, vizinho de Petya e excelente programador, ouviu os gritos e está disposto a ajudar. Como a polícia do queijo não permite a venda de cheddar para qualquer um, você fará um programa para facilitar as compras da querida mãe.

Todos os candidatos à compra de cheddar devem passar por um desafio de sorte. Eles recebem N ($0 < N \leq 1000$) dados (honestos, com 6 lados e valores de 1 a 6) e um número V ($0 \leq V \leq 6000$). Todos os N dados são lançados e o valor S é calculado sendo a soma dos números sorteados. Se S for maior ou igual a V , a compra está permitida, caso contrário, o comprador volta para casa de mãos vazias.

Seu programa deve calcular, dado N e V , a probabilidade da mãe de Petya conseguir comprar cheddar.

Entrada

Uma única linha com dois número inteiros, N e V , respectivamente.

Saída

Imprima uma única linha com um único número com precisão de quatro casas decimais, a probabilidade da mãe de Petya conseguir comprar cheddar.

Exemplo de entrada	Exemplo de saída
1 6	0.1667

Exemplo de entrada	Exemplo de saída
1 5	0.3333

Exemplo de entrada	Exemplo de saída
3 18	0.0046

Exemplo de entrada	Exemplo de saída
3 12	0.3750

B: Final Mundial de 2015

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

Preocupado com a atual situação de crise no transporte aéreo, o diretor regional do concurso do ICPC no Brasil já iniciou seus preparativos para fazer as reservas das passagens aéreas para as finais mundiais em Marrocos em 2015. O primeiro passo foi estudar a malha aérea disponível, em que cada voo tem um certo preço e liga duas cidades (estamos, na verdade, chamando de voo apenas um trecho *non stop* de um voo comercial). O objetivo do diretor é fazer várias consultas nesta malha de voos.

Em geral desejamos fazer voos sem escalas, mas estes podem ser muito caros. Para contornar este fato o diretor deseja permitir algumas escalas possíveis. Assim, ele ordenou as várias cidades da malha em sua ordem de preferência para fazer escala. Ou seja, a cidade de índice 1 é a que ele prefere fazer escala, seguida pela cidade 2, e assim por diante.

As consultas que o diretor fará são então do seguinte tipo: é dada a cidade de partida e de chegada e um número t de cidades em que o diretor permite que sejam feitas escalas. Seu programa deverá encontrar o custo de um voo de custo mínimo entre as cidades que faça, no máximo, escalas nestas cidades. Por exemplo, se $t = 1$ você deverá encontrar o custo de um voo de custo mínimo entre as duas cidades que seja ou *non stop* ou que faça uma escala na primeira cidade.

Entrada

A primeira linha consiste em dois inteiros n ($1 \leq n \leq 100$) e m ($1 \leq m \leq 100000$), indicando o número de cidades e o número de escalas. Nas m linhas seguintes temos três inteiros u , v e w ($1 \leq u, v \leq n$ e $0 \leq w \leq 100$) indicando que existe uma escala que vai de u para v com custo w . Em seguida um inteiro c ($1 \leq c \leq 10000$) indicando o número de consultas, e nas c linhas seguintes temos três inteiros o , d e t ($1 \leq o, d \leq n$ e $0 \leq t \leq n$) onde o é a cidade de origem, d é a cidade de destino e t indica que as cidades $1, 2, \dots, t$ podem ser usadas para escalas.

Saída

Para cada consulta, na ordem da entrada, você deve imprimir uma linha contendo o custo mínimo ou -1 caso não exista caminho entre as duas cidades obedecendo as restrições dadas.

Exemplo de entrada	Exemplo de saída
4 7	3
4 1 0	0
2 1 3	-1
1 4 20	
2 3 15	
4 2 1	
3 1 21	
1 2 0	
3	
2 1 0	
4 2 2	
4 3 1	

Exemplo de entrada	Exemplo de saída
5 10	-1
4 5 2	13
2 1 4	2
1 2 7	-1
2 4 7	
5 2 1	
4 1 2	
4 5 12	
5 4 4	
5 3 7	
3 5 9	
4	
2 5 0	
3 4 5	
4 5 1	
2 3 2	

C: Cola

Arquivo: cola.[c|cpp|java|pas]

Os alunos da quarta série do colégio Trale Lewous fizeram prova de Matemática no primeiro horário, todos entregaram uma folha apenas com as respostas e seus respectivos índices. Dona Maria Eduarda, a professora, suspeita Ricardinho de ter colado de Juan, o melhor aluno da turma.

Ricardinho foi chamado para conversar com a professora durante o recreio, a fim de verificar se houve cola ou não. Entretanto, ao ver que poderia ser pego, o menino, que é muito travesso, distraiu Maria Eduarda e apagou o índice das respostas na folha entregue. Por exemplo, se inicialmente estava escrito 1) 20 2) 30 3) 35 ..., agora restou apenas 20 30 35 ...

A prova consistia de N questões, todas com respostas numéricas. Entretanto, o menino conseguiu responder apenas M delas. A professora lembra, com certeza, que Ricardinho as escreveu em ordem na folha de respostas. Ou seja, se ele respondeu, por exemplo, as questões 2 e 4, a resposta da questão 2 está escrita antes da resposta da questão 4.

Tendo as N respostas de Juan, também em ordem, Maria Eduarda zerará a prova de Ricardinho se houver uma forma de casar o valor das M respostas do aluno malandro com M das N respostas do aluno caxias, sem desrespeitar a ordem delas.

Escreva um programa que, dado as N respostas de Juan e as M respostas de Ricardinho, diga se a prova deverá ser zerada.

Entrada

A primeira linha contém dois inteiros N e M ($1 \leq M \leq N \leq 10^5$). A segunda linha contém N inteiros s_i ($1 \leq s_i \leq 10^3$) descrevendo as respostas de Juan. A terceira linha contém M inteiros r_j ($1 \leq r_j \leq 10^3$) descrevendo as respostas de Ricardinho.

Saída

Imprima uma linha contendo `sim` se a prova deverá ser zerada ou `nao`, caso contrário.

Exemplo de entrada	Exemplo de saída
5 3 2 7 4 3 2 7 3 2	sim

Exemplo de entrada	Exemplo de saída
5 3 2 7 4 3 2 2 4 7	nao

D: Tesouro

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

Vasya e Petya adoram caça ao tesouro. O processo de diversão da dupla é agilizado por um site dedicado ao esporte, lá os amigos têm acesso às coordenadas do tesouro, direções e dicas de como encontrá-los.

Tendo anos de experiência e mais de 8192 tesouros encontrados, Vasya decidiu não apenas procurá-los, mas também escondê-los. Ambicioso, ele quer gerar uma experiência a ser lembrada o resto da vida para quem for atrás deles. O rapaz decidiu juntar seu conhecimento de geometria com o seu hobby para gerar um desafio formidável.

Os caçadores receberão a descrição de dois retângulos R_0 e R_1 . O tesouro está escondido na região de intersecção de R_0 e R_1 .

Como nem tudo na vida é fácil, existem regras para que o tesouro de Vasya seja listado no site. A área de busca não pode exceder $10m^2$. Por isto, Vasya pediu a Petya que verificasse a qualidade dos desafios. O amigo deve dizer se a área de busca tem tamanho adequado, se não há intersecção, se a área é muito grande, se é apenas uma linha ou se é apenas um ponto.

Tendo se formado em culinária, Petya não é muito bom em geometria. Ele admite não saber resolver o problema de seu amigo e está pedindo sua ajuda.

Entrada

Duas linhas, cada descrevendo um retângulo através quatro inteiros, duas coordenadas (em metros), X_0, Y_0, X_1 e Y_1 ($0 \leq X_0 < X_1 \leq 1000$ e $0 \leq Y_0 < Y_1 \leq 1000$), dois vértices opostos.

Saída

Imprima uma única linha com o resultado da avaliação da área de busca: "adequada", se estiver adequada e não for apenas ponto nem linha, "linha", se for uma linha, "ponto", se for apenas um ponto, "inexistente", se não existir, "grande", se for grande de mais.

Exemplo de entrada	Exemplo de saída
0 0 40 40 39 0 40 40	grande

Exemplo de entrada	Exemplo de saída
0 0 40 40 39 39 40 40	adequada

E: Surf Aquático

Arquivo: surf.[c|cpp|java|pas]

Este mês a Nlogônia está sediando a copa mundial de Surf Aquático. Neste esporte, um competidor fica N minutos dentro de um aquário fazendo acrobacias com sua prancha, tentando impressionar os juízes.

Ao contrário do Surf comum, os juízes dão uma nota ao competidor a cada minuto. Assim, um competidor recebe uma sequência de N notas a cada performance.

Uma *bateria* (de notas) é uma sequência contínua de B notas. O *resultado* de uma bateria é dada pela soma de todas as notas nela, exceto pela maior e pela menor nota, que são descartadas. Assim, o resultado da bateria (2, 3, 7, 5), por exemplo, é 8.

O *score* final do competidor é dado pela soma dos resultados de todas as baterias de tamanho B em sua sequência de N notas. Assim, se as notas são indexadas de 1 a N , soma-se o resultado da bateria contendo as notas de 1 a B , da bateria contendo as notas de 2 a $B + 1$, etc., até a bateria contendo as notas de $N - B + 1$ a N .

Dada a sequência de notas de um competidor, determine seu *score*.

Entrada

A primeira linha contém dois inteiros N e B ($3 \leq B \leq N \leq 2 \times 10^5$). A segunda linha contém N inteiros n_i ($1 \leq n_i \leq 100$), indicando, em ordem, a sequência de notas do competidor.

Saída

Imprima uma linha contendo o *score* do competidor.

Exemplo de entrada	Exemplo de saída
4 4 2 3 7 5	8

Exemplo de entrada	Exemplo de saída
7 5 2 8 3 4 8 1 2	39