VIII Simulado dos Ingressantes

Instituto de Matemática e Estatística

Universidade de São Paulo

Caderno de Problemas

Departamento de Ciência da Computação IME-USP
Sexta, 26 de Maio de 2023.

Instruções

- A competição tem duração de 5 horas;
- Os times são compostos de no máximo 3 integrantes;
- Os times podem utilizar somente um computador;
- Não é permitida a consulta de materiais durante a prova, exceto documentação das linguagens;
- O ambiente da prova será o CodeForces (www.codeforces.com).

- A entrada de cada problema deve ser lida da entrada padrão (teclado);
- A saída de cada problema deve ser escrita na saída padrão (tela);
- Siga o formato apresentado na descrição da saída, caso contrário não é garantido que seu código será aceito.

Vereditos das submissões	
In queue	Paciência
Accepted	Código aceito. Parabéns!
Wrong answer	Errado. Pode tentar novamente.
Time limit exceded	Seu programa demora muito para dar a resposta (certa ou errada). Pode tentar novamente.
Runtime error	Erro em tempo de execução (ex.: *segmentation fault*). Pode tentar novamente.
Compilation Error	Erro de compilação. Pode tentar novamente.

Problem A. Ágoranegócio

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Rodingo "Cria", o deus da Criação, convidou os semideuses Renang e Wan para passarem o break no Monte Olimpo. Como os dois gostam muito de frutas, eles irão visitar os t mercados da região grega, conhecidos pela qualidade de seus caquis, bananas-da-terra e outras iguarias típicas.

Quando Renang e Wan entram em um mercado, eles visitam cada uma das n vendas que há no mercado e compram exatamente um pacote de frutas de cada uma delas. Cada vendedor do mercado vende suas frutas em pacotes com quantidades distintas de frutas e, para facilitar as escolhas dos ilustres clientes, os pacotes estão ordenados em ordem crescente sobre as mesa de cada venda.

Após sair do mercado, para satisfazer um capricho de Rodingo, Renang e Wan querem abrir os pacotes e distribuir todas as frutas entre os dois de forma que cada um fique com uma quantidade igual de frutas ao final da distribuição.

O problema é que nenhum dos três personagens é muito bom com contas. Por isso é seu trabalho ajudá-los a decidir se, para cada um dos mercados visitados, a tarefa é possível ou não.

Input

A primeira linha da entrada é composta de um único inteiro t ($1 \leq t \leq 100$), o número de mercados do Monte Olimpo. A seguir, cada mercado é descrito.

A primeira linha de cada mercado é um único inteiro n ($1 \leq n \leq 10^5$), o número de vendas do mercado. A seguir, cada uma das n vendas é descrita.

A primeira linha da descrição de cada venda é a quantidade m ($1 \leq m \leq 10^5$) de pacotes de frutas à venda. A segunda linha da descrição de cada venda contém m inteiros positivos distintos $a_1 < a_2 < \cdots < a_m$ ($1 \leq a_j \leq 10^9$, para $j = 1, 2, \ldots, m$), separados por espaço, onde a_i é a quantidade de frutas que o i-ésimo pacote sobre a mesa dessa venda contém.

É garantido que a soma do número de pacotes em todos os mercados é no máximo 10^5.

Output

Para cada mercado, imprimir em uma única linha "SIM" (sem aspas) se Renang e Wan conseguirão fazer as compras e a divisão do jeito desejado e "NAO" (sem aspas) caso isso seja impossível.

Example

standard input	standard output
2 2 1 1 2 2 4 1 3 2 5 8	NAO SIM

Note

Há 2 mercados.

No primeiro mercado, há 2 vendas: a primeira delas vende apenas 1 pacote com 1 fruta e a segunda vende 2 pacotes com 2 e 4 frutas, respectivamente. É possível provar que, independente das escolhas de Renang e Wan, a distribuição será impossível.

No segundo mercado, há somente 1 venda, que vende pacotes com 2, 5 e 8 frutas, respectivamente. Renang e Wan podem satisfazer a vontade de Rodingo (por exemplo comprando o pacote com 8 frutas e cada um deles ficará com 4 frutas após a distribuição igualitária).

Problem B. Balões Quânticos

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Byakronos estava organizando o primeiro contest da maratona. Entretanto, perto do começo da prova, ele percebeu que não haviam balões suficientes!

Por isso, Byakronos mandou Mori buscar balões com a charrete do MaratonUSP. Felizmente, Mori se deparou com a loja do Deus Hélio que vende balões "quânticos". Assim, Mori comprou balões **inflados** (para economizar tempo!) de N cores, em que b_i balões da cor i ($1 \leq i \leq N$) foram comprados.

Ao colocar os balões no caminhão, Mori percebeu que a charrete comportava no máximo S balões inflados. Desse modo, Mori precisa esvaziar balões de modo que restem no máximo S balões inflados.

Como os balões são quânticos, eles só esvaziam se Mori olhar para eles. Ainda mais, os balões inflados obedecem às regras de decaimento radioativo, tendo uma meia-vida de 1 segundo. Em outras palavras, a cada 1 segundo que Mori olha para x balões inflados, $\lfloor \frac{x}{2} \rfloor$ continuarão inflados no fim desse segundo.

Assim, para esvaziar os balões, Mori irá a cada segundo escolher uma cor i e olhar para os balões dessa cor por exatamente 1 segundo (caso Mori olhasse para mais de uma cor, ele iria se perder na conta!).

Obviamente não é saudável olhar balões quânticos por muito tempo. Então Mori se pergunta, qual é o número mínimo de segundos necessários para que restem no máximo S balões inflados?

Input

O input consiste de 2 linhas.

A primeira linha contém dois inteiros N, S ($1 \leq N \leq 10^5, 0 \leq S \leq 10^9$) separados por espaço — representando o número de cores de balões e o máximo de balões inflados que o caminhão comporta, respectivamente.

A segunda linha contém N inteiros $b_1, b_2, \ldots, b_N$ ($1 \leq b_i \leq 10^9$) — representando a quantidade de balões de cada cor comprados.

Output

Imprima um inteiro, o número mínimo de segundos necessários para que restem no máximo S balões inflados.

Examples

standard input	standard output
2 5 3 5	1
5 0 1 1 1 1 1	5

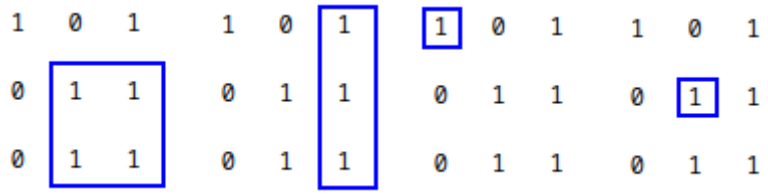
Problem C. Convite dos Deuses

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Os membros do MaratonUSP foram convidados pelo deus Zeos para visitar o Olimpo. May, sabendo dos banquetes lendários que lá ocorrem, logo se prontificou.

Após um longo passeio pelos campos de Caieffes, os membros do MaratonUSP foram convidados para um banquete. Contudo, para a surpresa de quase todos, o banquete consistia somente de coxinhas e bolinhas de queijo! Secretamente, Thilio, o fresco, disse aos deuses que esse era o único alimento que os membros do MaratonUSP gostam, para que não tivesse que lidar com nenhum alimento da Grécia antiga.

May ficou revoltada, pois odeia coxinhas. Os deuses serviram os salgadinhos em uma mesa, organizados em n linhas e m colunas. Como May ainda quer visitar mais locais antes de voltar ao IME, ela não tem tempo de banquetear com seus colegas. Ela decidiu ir à mesa de salgadinhos e pegar todos os salgadinhos de uma subseção da mesa de uma só vez. Para facilitar, ela vai selecionar uma subseção retangular da mesa (isto é, um subconjunto contínuo de linhas e colunas) e pegar **todos** os salgadinhos desta subseção. A seguinte imagem contém algumas escolhas válidas para o terceiro caso teste.



Ajude May a descobrir o número máximo de bolinhas de queijo que pode pegar dessa forma **sem pegar nenhuma coxinha**.

Input

A primeira linha contém dois inteiros positivos n e m separados por um espaço, as dimensões da mesa de salgadinhos. É garantido que $n \times m \leq 400$.

As próximas n linhas contém m números $a_{i,j} \in \{0, 1\}$ separados por espaços. Se $a_{i,j} = 0$ indica que o salgadinho na linha i e coluna j é uma coxinha. Se $a_{i,j} = 1$ é uma bolinha de queijo.

Output

Um inteiro, o número máximo de bolinhas de queijo que May consegue pegar.

Examples

standard input	standard output
2 2 1 1 1 1	4
4 4 1 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1	4
3 3 1 0 1 0 1 1 0 1 1	4

Note

Para o primeiro exemplo, May pode selecionar as colunas 1-2 e linhas 1-2.

Para o segundo exemplo, May tem varias opções.

- linhas 1-2 e colunas 1-2;
- linha 1 e colunas 1-4;
- linha 4 e colunas 1-4;
- linhas 1-4 e coluna 1.

Problem D. Dupla Sertaneja do Olimpo

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Henrique e Juliana foram uma dupla sertaneja do Panteão Olímpico Lírico dos Imortais (POLI), cujo objetivo era criar **música divina** para honrar os deuses. A partitura de uma **música divina** é uma sequência de notas representadas apenas pelos números 0, 1 e 2 e é tal que a soma de quaisquer três números consecutivos é exatamente 3.

Recentemente, foi achada uma antiga partitura da famosa dupla, mas algumas das suas notas estavam borradas. Assim, o trabalho de recuperar a partitura original foi encomendado a Pedro, o músico. Pedro teme que esse trabalho não seja possível, pois dependendo das notas borradas, pode existir mais de uma **música divina** cuja partitura (com algumas notas borradas) seja igual à que foi achada.

Atormentado por essa possibilidade, Pedro pediu para vocês contar o número de distintas partituras de **músicas divinas** que poderiam ter sido a originalmente escrita pela dupla sertaneja.

Input

A primeira linha da entrada contem um único inteiro n ($3 \leq n \leq 10^6$), a quantidade de notas na partitura achada.

A segunda linha contem n inteiros d_i ($-1 \leq d_i \leq 2$ para todo $1 \leq i \leq n$), as notas da partitura. O inteiro -1 representa uma nota borrada.

Output

Imprima um único inteiro: a quantidade de partituras de **músicas divinas** que poderiam ter sido a originalmente escrita por Henrique e Juliana.

Examples

standard input	standard output
5 0 -1 1 -1 2	1
5 -1 -1 2 -1 -1	2

Problem E. Enigma da Esfinge

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Otávio, o arqueólogo, estava prestes a entrar no sítio arqueológico onde atualmente faz sua pesquisa quando foi parado pela esfinge conhecida como Matheusfinge.

Para que pudesse passar e continuar sua pesquisa Otávio precisaria responder o enigma da esfinge. Dessa vez, Matheusfinge fez uma pergunta simples: "Qual porcentagem das estatísticas são falsas?"

A pergunta pegou Otávio de surpresa e ele imediatamente respondeu "75%", o que estava completamente errado. Felizmente as esfinges trabalham retroativamente, ou seja, se por algum motivo essa resposta passar a ser correta a esfinge vai liberar a passagem de Otávio.

Otávio passou sua vida inteira estudando arqueologia, então ele sabe exatamente o número de estatísticas verdadeiras v e o número de estatísticas falsas f.

Outra coisa que Otávio aprendeu em sua pesquisa é que proclamar a frase "Mais que 75% das estatísticas são falsas" conta como uma estatística e acaba mudando a quantidade de estatísticas verdadeiras ou falsas de acordo com v ou f. Se $\frac{f}{v+f} > 75\%$ e essa frase for proclamada, ela é verdadeira, e logo o valor v aumenta em 1. Caso contrário, o valor de f aumentaria em 1.

Otávio precisa voltar ao trabalho e pediu a sua ajuda para escrever um programa que dado o número de estatísticas verdadeiras v e o número de estatísticas falsas f mostra para ele quantas vezes a frase "Mais que 75% das estatísticas são falsas" precisa ser usada para que exatamente 75% das estatísticas sejam falsas.

Input

A entrada contêm multiplos casos de teste.

A primeira linha do input contêm um número $1 \leq t \leq 10^5$, a quantidade de testes.

Cada uma das próximas t linhas contêm dois números v e f ($0 \leq v, f \leq 10^8$), o número de estatísticas verdadeiras e o número de estatísticas falsas, respectivamente. É garantido que $v+f > 0$ e que $\frac{f}{v+f} \neq 0.75$.

Output

A saída deve conter t números, um para cada teste. O i-ésimo número deve ser a quantidade de vezes que Otávio precisa falar "Mais que 75% das estatísticas são falsas" para que exatamente 75% das estatísticas sejam falsas no i-ésimo caso de teste.

Examples

standard input	standard output
1 1 2	1
1 0 1	3

Problem F. Fraude na votação

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Na busca por adoração e aumento do ego, os deuses do Olimpo organizam uma votação na Terra, a cada 100 anos, para determinar o deus mais adorado pelos humanos. Nesta edição, m deuses, incluindo Byakronos, Marceloki e Zeos, os favoritos, estão competindo pelo título. Apenas algumas pessoas aleatoriamente selecionadas são convidadas para votar.

A profeta Kobus, seguidora fervorosa de Zeos, está determinada a garantir sua vitória. Ela possui conhecimento privilegiado sobre os eleitores e sabe exatamente quantos votos cada deus tem até o momento. Para o i-ésimo deus, ela tem a informação de que ele recebeu a quantidade parcial de votos a_i.

No entanto, Kobus precisa da sua ajuda para determinar a menor quantidade t de votos que ela deve comprar para assegurar a vitória de Zeos. Em outras palavras, Zeos deve ter mais votos do que qualquer outro deus na competição. É importante observar que Zeos é o primeiro deus na lista fornecida por Kobus e que ao comprar um voto, este é retirado de outro deus e transferido para Zeos.

Input

Na primeira linha é dado um inteiro m ($3 \leq m \leq 2 \cdot 10^5$) - o número de deuses concorrendo ao título.

Na segunda linha são dados m inteiros $a_1, a_2, \ldots, a_m$ ($1 \leq a_i \leq 10^9$) - onde a_i indica quantos humanos votaram no i-ésimo deus.

Output

A saída deve ser um único inteiro t indicando o número mínimo de votos que Kobus deve comprar.

Examples

standard input	standard output
3 1 1 7	4
3 1 2 6	3
6 2 4 2 5 6 5	4

Note

No primeiro examplo, é possível provar que 3 votos não são suficientes pois, independente de como Kobus os comprasse, o terceiro candidato não teria menos votos que Zeos.

Por outro lado, se ela comprar 4 votos do terceiro candidato para Zeos, ele passaria a ter 5 eleitores (enquanto o terceiro candidato, apenas 3).

Problem G. Gusteseu e a Maynotauro

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

A antiga e movimentada cidade de Cnossos era conhecida pelo seu misterioso labirinto, que era composto por um conjunto de $N \cdot M$ salas quadradas, dispostas na forma de um grid N por M. As suas salas eram nomeadas por um par ordenado (A, B), em que A é a linha e B é a coluna em que essa sala se encontrava. A entrada do labirinto se localizava na posição $(1, 1)$.

Nessa cidade, havia um destemido aventureiro chamado Gusteseu. Ele era conhecido por sua coragem e habilidade em desvendar os enigmas mais complexos.

Certo dia, Gusteseu recebeu uma missão extraordinária: explorar o labirinto e encontrar um tesouro lendário que se dizia estar escondido em suas profundezas. No entanto, havia um desafio assustador no caminho de Gusteseu. O labirinto era habitado por uma criatura temida por todos: a poderosa Maynotauro.

A Maynotauro, uma criatura meio mulher e meio touro, guardava a entrada do labirinto e protegia o caminho para o tesouro. Sua morada era conhecida como a sala da Maynotauro, localizada na posição (X, Y) do labirinto.

Determinado a completar sua missão, Gusteseu sabia que teria de evitar a sala da Maynotauro. Ele entendia que qualquer passagem por essa sala resultaria em sua captura e fracasso na busca pelo tesouro.

Como o clima dentro do labirinto era muito desagradável, Gusteseu queria chegar à sala do tesouro de forma eficiente, passando pelo menor número de salas possível. Assim, ele desconsiderava quaisquer outras formas de se fazer tal percurso.

Com sua mente afiada e conhecimentos em programação competitiva, Gusteseu decidiu usar suas habilidades para calcular quantas formas eficientes existiam de se chegar à sala do tesouro, localizada na posição (N, M), sem passar pela sala da Maynotauro.

Input

A entrada consiste de uma única linha com quatro inteiros separados por espaços, N, M $(1 \leq N, M \leq 30)$, X $(1 \leq X \leq N)$ e Y $(1 \leq Y \leq M)$.

N e M indicam o tamanho do labirinto e a posição (N, M) da sala do tesouro.

X e Y indicam a posição (X, Y) da sala da Maynotauro.

Output

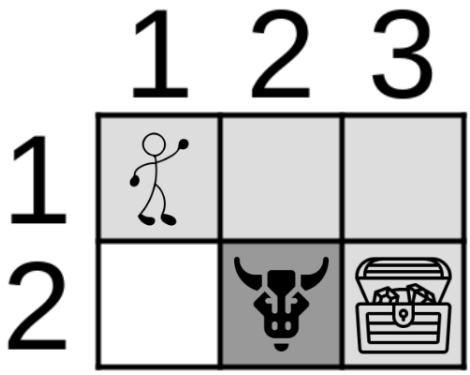
Imprima um único número indicando o número de formas eficientes distintas de se chegar à sala do tesouro que foi calculado por Gusteseu.

Examples

standard input	standard output
2 3 2 2	1
4 4 3 2	11
30 30 15 15	23844478996709040

Note

Exemplo do caso de teste 1:



Só há uma forma válida, que está destacada.

Problem H. Herói Morethor

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Segundo vestígios históricos, acredita-se que na antiga Nlogônia todos os cidadãos viviam em paz e harmonia, confiantes de que estavam protegidos de qualquer inimigo graças a um grande herói: Morethor. O deus Hadesoko, conhecido por seus poderosos monstros, buscava atormentar constantemente a humanidade, mas nosso grande herói impedia seus feitos maléficos. Entretanto, recentes descobertas apontam para um período de trevas: após o encontro de Morethor com o deus Promethilio, nosso herói ficou um longo tempo desaparecido, permitindo que Hadesoko invadisse as terras de nossos ancestrais. Sabemos que Morethor reapareceu em algum momento do passado e, sem tempo para treinar, imediatamente começou a caçar pelos monstros enviados por Hadesoko e a enfrentá-los na ordem em que os encontrava. Entretanto, apesar dos dados coletados, os pesquisadores ainda não foram capazes de concluir se Morethor derrotou todas as forças do mal ou se este falhou em sua missão. Assim, eles pediram sua ajuda para avaliar os dados e sanar essa grande dúvida!

Dada a habilidade H de Morethor logo após seu retorno, determine se este é capaz de enfrentar todos os N monstros de Hadesoko. O i-ésimo monstro encontrado possui uma força F_i. Morethor enfrenta um único monstro por vez e é capaz de derrotá-lo se sua habilidade atual é maior ou igual à força do monstro. Caso nosso herói vença a luta, adquire a força de seu oponente (ou seja, soma à sua habilidade a força do monstro). Caso contrário, é derrotado.

Input

A primeira linha da entrada contém um inteiro N ($0 \leq N \leq 10^5$) e um inteiro H ($0 \leq H \leq 10^4$) — o número de monstros e a habilidade inicial do herói, respectivamente.

A segunda linha contém N inteiros F_i ($0 \leq F_i \leq 10^4$) — a força do i-ésimo monstro encontrado por Morethor.

Output

Imprima "SIM" (sem aspas) se o herói é capaz de enfrentar todos os monstros ou "NAO" (sem aspas) caso contrário.

Examples

standard input	standard output
5 10 1 2 3 4 5	SIM
4 1 1 2 4 8	SIM
4 7 1 1 1 11	NAO

Note

No segundo caso de teste, o herói começa com habilidade igual à 1, sendo assim capaz de vencer o primeiro monstro e passando a ter habilidade 2. Na segunda batalha, também é capaz de vencer o monstro e passa a ter habilidade 4. Na terceira e quarta batalha, o mesmo acontece, sendo nosso herói capaz de enfrentar todos os monstros.

Problem I. Ideias Primordiais

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Antes de começarmos a contar essa magnífica história é preciso algumas definições:

- Dizemos que um vetor $\mathcal{P}$ é uma permutação se cada uma das 26 letras do alfabeto aparece em exatamente uma posição de $\mathcal{P}$. Por exemplo, o vetor $\mathcal{A} = (D, F, C, O, I, L, S, W, V, X, H, K, A, T, Q, M, R, Y, Z, J, B, E, U, N, G, P)$ é uma permutação, enquanto o vetor $\mathcal{B} = (A, F, C, O, I, L, S, W, V, X, H, K, A, T, Q, M, R, Y, Z, J, B, E, U, N, G, A)$ não, pois a letra 'A' aparece múltiplas vezes.
- Seja L a ℓ-ésima letra do alfabeto e $\mathcal{P}$ uma permutação, então definimos $\mathcal{P}(L)$ como sendo a ℓ-ésima letra de $\mathcal{P}$. Usando o exemplo do item anterior $\mathcal{A}(\text{D}) = \text{O}$, porque D é a quarta letra do alfabeto e O é a quarta letra de $\mathcal{A}$. Similarmente, dado um texto T e uma permutação $\mathcal{P}$, definimos $\mathcal{P}(T)$ o texto resultante de aplicar $\mathcal{P}$ para cada letra em T. Por exemplo, $\mathcal{A}(\text{EU AMO AC}) = \text{IB DAQ DC}$.
- Dado um conjunto de palavras U dizemos que um texto $T \in U*$ se T contém somente palavras de U. Por exemplo, se $U = \{AC, SIM, CORRETO\}$ então o texto *SIM SIM AC* pertence a $U*$, enquanto o texto *AC WA AC* não.

Contaremos agora a história da origem da programação competitiva entre os humanos. Diz a lenda que há milhares de anos, na antiga Nlogônia, o deus Rárada constantemente inventava novos problemas e novos algoritmos para resolvê-los, e compartilhava essa ideia entre os deuses. Sabe-se que os deuses usavam pouquíssimas palavras quando se comunicavam entre eles, mais especificamente, as divindades só utilizavam palavras dentro do conjunto *U = {AC, AMOR, BRASILEIRO, CAVALO, MARATONUSP, OSSO, OVO, PATA, RARADA, TLE, VOO}*; não questione o porquê eles escolheram essas palavras. Entre esses deuses estava Promethilio, que maravilhado com essas novas ideias, acreditava que esse conhecimento deveria ser passado aos humanos. Contudo, temendo que os humanos fossem se tornar muito poderosos ele não falava exatamente a mesma coisa que Rárada. Toda vez que Promethilio escutava uma frase $T \in U*$ vinda de Rárada, ele escolhia uma permutação $\mathcal{P}$ e enviava o texto $\mathcal{P}(T)$ para o herói Morethor (talvez você já tenha escutado falar dele em outra questão). Morethor então transcrevia a frase $\mathcal{P}(T)$ em tábulas de pedra com o intuito de disseminar essas ideias e de preservar os conhecimentos de progamação competitiva para sempre.

Otávio, o arqueólogo, encontrou uma tábula de pedra com um texto S e está extremamente empolgado com a possibilidade de ter encontrado uma das antigas tábulas escritas por Morethor com as ideias primordiais de Rárada. Ele agora precisa da sua ajuda para decidir se ele realmente encontrou uma relíquia histórica ou somente um texto qualquer.

Input

O input contém duas linhas, a primeira contém um inteiro N. A segunda linha contém o texto S escrito na tábula. O texto contém exatamente N palavras e $1 \leq |S| \leq 10^6$, ou seja, o texto tem entre 1 e 1000000 de caracteres. Todos os caracteres são letras maiúsculas ou espaço em branco. Todas as palavras são separadas por exatamente um espaço.

Output

Se existe um texto $T \in U*$ e uma permutação $\mathcal{P}$ tal que $\mathcal{P}(T) = S$ a saída deverá conter duas linhas: na primeira imprima "Y", e na segunda a permutação $\mathcal{P}$ no formato "$\mathcal{P}[1]\mathcal{P}[2]...\mathcal{P}[26]$". Se para qualquer texto $T \in U*$ e qualquer permutação $\mathcal{P}$ temos $\mathcal{P}(T) \neq S$ então a saída deverá ser somente uma linha contendo "N".

Examples

standard input	standard output
3 NBSBUPOVTQ PWP BD	Y BCDEFGHIJKLMNOPQRSTUVWXYZA
2 BD CMOR	N
6 PSDETVXTSQ DHQS SDSDRD QEEQ ADKD DG	Y DPGRXBCFTIJVHLQAMSEKNOUWYZ
2 OSSO PATA	Y ABCDEFGHIJKLMNOPQRSTUVWXYZ

Note

No primeiro caso teste, a frase dita por Rárada foi MARATONUSP OVO AC. No segundo caso teste, não existe permutação que levaria uma frase dita por Rárada ao texto da entrada No terceiro caso teste, a frase dita por Rárada foi BRASILEIRO AMOR RARADA OSSO PATA AC.

Para os casos positivos note que mais de uma permutação poderia ser aceita como resposta.

Problem J. Jantar dos Deuses

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Em ocasiões especiais, os deuses se reúnem para um divino jantar no bandejão. Anthenas sempre convida seu colega Nathenas para esses encontros, mas Nathenas, em uma demonstração de extrema arrogância, constantemente se recusa a participar.

Como a sua presença nos jantares é extremamente rara, rezam as lendas que, desde que emergiu da cabeça de Zeos, Nathenas recarregou seu saldo no bandejão apenas uma vez. Sabe-se que cada recarga pode ser de no máximo 200 dracmas (sim, os deuses ainda usam essa moeda), sendo que cada refeição custa exatamente duas dracmas.

A partir disso, dado o valor da recarga feita por Nathenas e o seu saldo atual, descubra quantas refeições ele fez no bandejão.

Input

A entrada consiste em uma única linha com dois inteiros x e y separados por espaço, respectivamente o valor da recarga e o saldo atual de Nathenas. É garantido que $0 \leq y \leq x \leq 200$ e que x e y ambos têm a mesma paridade.

Output

A saída deve conter um único inteiro, a quantidade de vezes que Nathenas comeu no bandejão.

Examples

standard input	standard output
50 34	8
131 47	42