Problem A. Astronautas em Apuros

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

Um grupo de astronautas estavam explorando o longínquo planeta de Tribur, quando uma tempestade violenta danificou sua nave e os deixou a deriva.

O astronauta Gasparinov foi escolhido para ser enviado na missão de resgate. Ele deve pousar sua nave no planeta, desacoplar seu veículo, dirigir-se aos astronautas perdidos e retornar com eles em segurança para sua nave principal.

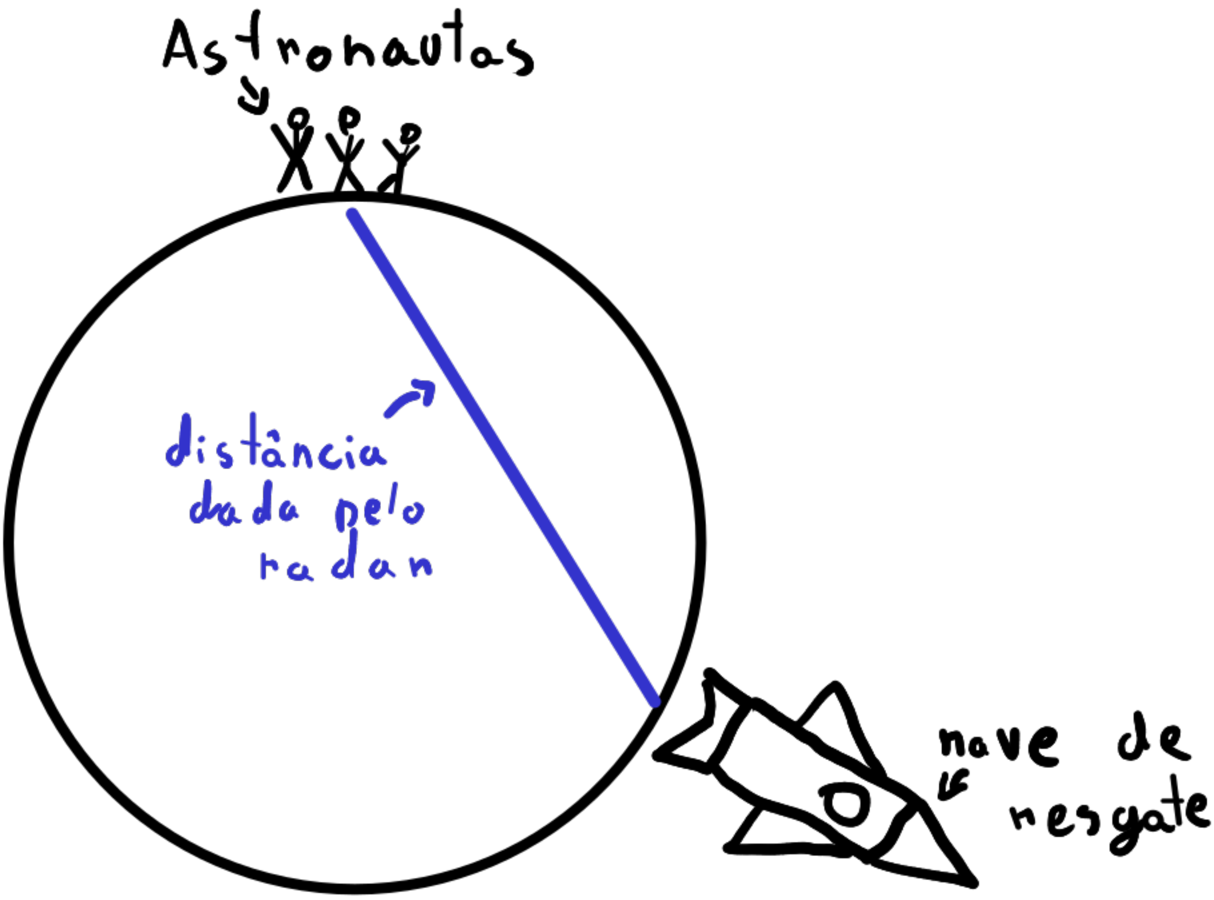
Durante o pouso de Gasparinov outra tempestade assolava o planeta e por esse motivo ele perdeu o controle durante o pouso.

Gasparinov conseguiu guardar a direção de onde estão os outros astronautas, porém não sabe o quão longe estão e teme que não haja combustível suficiente em seu veículo para buscá-los e retornar a nave.

Felizmente Gaparinov achou uma ferramenta capaz de indicar a distância absoluta em quilômetros entre as duas naves, porém ela é dada em linha reta (lembrando que o planeta Tribur tem o formato esférico) e Gasparinov só pode percorrer pela superfície do planeta.

O veículo consome 1 litro de combustível por quilômetro e possui um tanque acoplado.

Determine se há ou não combustível suficiente para a jornada de **IDA E VOLTA**.



Input

A entrada consiste de três inteiros R, D e C indicando respectivamente o raio do planeta, a distância em quilômetros indicada pelo aparelho e a quantidade de litros de combustível disponíveis. (É garantido que a entrada é válida).

- $1 \leq R, D, C \leq 10^9$.

Output

Você deve imprimir "possivel"(sem aspas) caso haja combustível suficiente e "impossivel"(sem aspas) caso não haja.

Examples

standard input	standard output
25 50 200	possivel
5000 300 1	impossivel
6 10 24	possivel

Problem B. Barata

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

A Barata diz que tem N saias de filó

É mentira da barata. Ela tem é M só

Há, há, há, hó, hó, hó

Ela tem é M só

Input

A entrada consiste de duas linhas.

A primeira linha contem o inteiro N.

A segunda linha contem o inteiro M.

- $0 \leq N, M \leq 10^5, N \neq M$

Output

Imprima um único inteiro, quantas saias tem a barata?

Example

standard input	standard output
7 1	1

Problem C. Cédulas Fracionárias

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

Joãozinho III é atualmente o rei da Joãolândia. Durante toda a vida ele nunca gostou de carregar moedas no bolso e resolveu acabar com elas de uma vez por todas. Seu plano é criar cédulas com valores fracionários para substituir as moedas antigas e já acionou a casa da moeda para tratar do assunto. O rei também exigiu que na nota fosse impresso o valor do numerador e do denominador para indicar o valor da cédula.

A casa da Moeda de Joãolândia começará a produção das novas cédulas em breve, porém para evitar desperdícios serão produzidos apenas células com valores maiores que zero e menores que 1, onde as frações já estejam simplificadas (gcd(numerador, denominador)=1).

O rei ressaltou que é muito supersticioso e pediu que o denominador da fração não exceda o seu número da sorte. A casa da Moeda precisa saber quantas cédulas novas serão criados.

Input

A entrada consiste de um inteiro D indicando o número da sorte do rei.

- $1 \leq D \leq 2 * 10^6$.

Output

Você deve imprimir o número de cédulas que entrarão em circulação.

Note que esse valor pode ser muito grande, então imprima ele em MOD $10^9 + 7$.

Examples

standard input	standard output
4	5
10	31

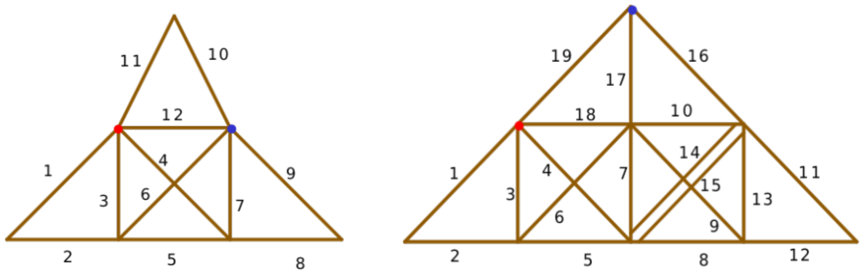
Problem D. Duende Entediado 2

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

Após se livrar do tédio na fábrica de presentes de natal (cf. Contest de Natal do BRUTE no codeforces depois da maratona), o duende inventou um novo modelo de negócios para gerar renda enquanto a fábrica está parada: construir casas com palitos de madeira.

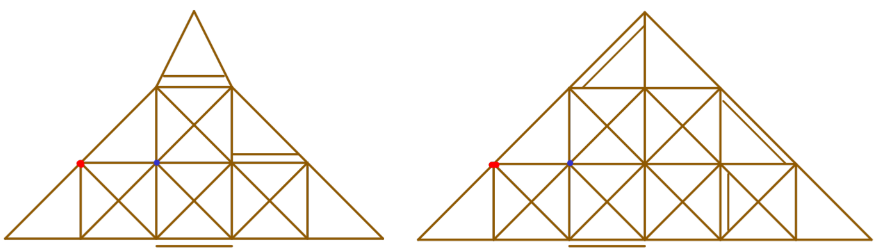
O duende possui atualmente diversas encomendas, porque seu trabalho é de alta qualidade e ele utiliza uma técnica diferente que ele aprendeu durante seu tédio: depois de colocar um palito de madeira em um ponto de um grid, o ponto inicial do palito seguinte sempre será o ponto final do palito anterior, seguindo esse processo até que a casa esteja construída. O ponto inicial do primeiro palito pode estar em qualquer lugar, mas o ponto final do último palito deve ser o ponto inicial ou o ponto final de um palito já inserido, e o duende sempre tentará minimizar a quantidade de palitos utilizados.

As casas possuem todas um formato triangular, mas elas podem ter bases de tamanho diferentes que, consequentemente, aumentam o tamanho da altura também. A ilustração a seguir mostra casas com bases de tamanho 1, 2, 3 e 4, bem como mostra o caminho utilizado para consumir o menor número de palitos nas bases de tamanho 1 e 2, com o número ao lado dos palitos indicando qual a ordem de colocação dos mesmos. Os pontos vermelhos nas ilustrações representam os pontos iniciais das construções e os pontos azuis representam os pontos finais das construções.



Casas de tamanho 1 e 2

Note que dois palitos podem compartilhar pontos iniciais e finais, como no caso dos palitos 14 e 15 na base de tamanho 2.



Casas de tamanho 3 e 4

Os palitos utilizados na construção das casas são especiais e só podem ser comprados em Nlogônia. Por isso, o duende quer aproveitar seu passeio por lá e comprar os palitos necessários para construir suas encomendas. Por isso, ele precisa da sua ajuda. Dado o número de palitos que o duende já possui e a lista de encomendas, ele pede que você calcule a quantidade de palitos que ele deve comprar em sua passagem por Nlogônia.

Input

Na primeira linha, serão fornecidos dois inteiros N e P, representando respectivamente o número de encomendas e o número de palitos que o duende tem atualmente. Na segunda linha, serão dados N inteiros a_i, com $i \in [1, N]$, que representam qual o tamanho da base da encomenda i

Limites:

- $1 \leq N \leq 2 * 10^5$
- $0 \leq P \leq 10^9$
- $1 \leq a_i \leq 2 * 10^5$

Output

Imprima o número de palitos que o duende precisará comprar para conseguir finalizar todas as encomendas.

Example

standard input	standard output
4 40 2 4 3 1	66

Problem E. Exames

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

Nos finais de semestre da UDESC, é muito comum circular entre os alunos uma tabela onde, dada uma média abaixo de 7.0 em uma disciplina, mostra quanto será necessário tirar no exame para conseguir ser aprovado na mesma disciplina.

Alguns alunos, porém, já estão cansados de terem que pedir por essa tabela ou de terem que olhar nessa tabela. Portanto, eles pediram para que você criasse um programa que calculasse o valor dessa tabela dado uma média abaixo de 7.0, lembrando que a fórmula do cálculo da média final é:

$$M_F = \frac{M_S * 6 + N_E * 4}{10}$$

Onde M_F é a média final, M_S é a média semestral e N_E é a nota do exame. Além disso, um aluno é aprovado na disciplina se $M_F \geq 5.0$, porém o Siga só considera notas com 1 casa decimal.

Sua tarefa é, dado um M_S, encontrar o menor valor com uma casa decimal para N_E tal que o aluno seja aprovado na disciplina. Caso o N_E encontrado possuir mais do que 1 casa decimal, N_E deve ser arredondado para baixo. Por exemplo, se $N_E = 2.15$, a resposta final deve ser $N_E = 2.1$.

Input

A entrada consiste de vários casos de testes.

Na primeira linha, será fornecido um inteiro T, o número de casos de testes.

Nas próximas T linhas, será fornecido um valor com uma casa decimal N, a média semestral do aluno na disciplina.

Limites:

- $1 \leq T \leq 100$
- $1.7 \leq N \leq 6.9$

Output

Para cada caso de teste, imprima um valor com uma casa decimal para N_E que satisfaça as condições do problema.

Example

standard input	standard output
3 5.0 1.7 6.9	5.0 9.9 2.1

Problem F. Família

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

A família Mews é muito grande, muito velha e muito amiga.

A família também se preocupa bastante com a qualidade de vida dos seus membros e todos sabem que, quanto maior o salário, maior a qualidade de vida. Por isso os Silva foram instruídos desde pequenos a sempre trocar de emprego se o salario ofertado pelo novo emprego for maior.

Como a família é muito amiga, os filhos sempre tentam puxar os seus ancestrais mais recentes para o mesmo emprego, ou seja, pai, avô e assim por diante.

Portanto, uma oferta de emprego pode ser representada por três valores, u, k e w, significando que o membro u e seus $k-1$ ancestrais receberam a oferta com salario w. Cada membro escolhe individualmente se aceita a oferta ou não.

Recentemente surgiu uma intriga na família, e os familiares estão comparando qual das subfamílias é mais rica.

Dizemos que a riqueza da subfamília de p é o somatório dos salários de todos os membros que são filhos diretos ou indiretos de p.

Adam é o patriarca da família e, portanto, sabe tudo sobre sua família. Sabe que a família possui n membros, sabe o pai de cada membro, o salario inicial e todos as ofertas de emprego recebidas recentemente pelos membros da família.

Infelizmente ele já está muito velho e precisa de sua ajuda. Adam vai te passar as informações e, ao longo do processo, perguntar qual a riqueza de determinada subfamília naquele momento.

Input

A primeira linha contem dois inteiros n e q separados por espaço.

A segunda linha contem $n-1$ inteiros p_i, cada um representando o pai do membro $i+1$. Adam é sempre o membro 1.

A terceira linha contem n inteiros w_i, cada um representando o salario do membro i antes dele receber ofertas de emprego.

Seguem então q linhas, cada linha começa com um inteiro o.

- Se $o = 0$, seguem três inteiros u, k e w, representando um oferta de emprego.
- Se $o = 1$, segue um inteiro u, representando uma pergunta de Adam. Qual a riqueza da subfamília de u?

Restrições:

- $1 \leq n, q \leq 10^5$
- $1 \leq p_i, u, k \leq n$
- $0 \leq w_i, w \leq 10^9$

Output

Para cada pergunta de Adam, imprima um único inteiro representando qual a riqueza da subfamília questionada.

Example

standard input	standard output
10 5 1 1 1 8 8 6 4 6 9 255 546 392 210 383 350 1130 334 64 360 1 3 0 2 1 350 1 8 0 9 4 500 1 1	392 2621 5066

Problem G. Goose

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

Em 3 de março de 1969, a Marinha dos Estados Unidos estabeleceu uma escola de elite para os melhores um por cento de seus programadores.

Seu objetivo era ensinar a arte perdida dos algoritmos e garantir que os homens que se formarem sejam os melhores programadores do mundo.

Eles conseguiram.

Hoje, a Marinha chama esse projeto de *B-plan team of Research of the United states Technicians of Encoding.*

O mundo chama de:

TOP CODE

Você é um desses alunos. A sua mais nova missão consiste em realizar dois milagres.

O mapa da região é representado por um mapa N por M onde cada célula do mapa tem um valor v_{ij}, representando a infraestrutura inimiga instalada, esse valor é negativo caso a infraestrutura seja aliada.

O TOP CODE destruirá um retângulo desse mapa, cabe a você determinar os dois pontos do retângulo com a maior soma de infraestrutura.

Input

A primeira linha consiste de dois inteiros N e M, separados por espaço.

Seguem N linhas com M inteiros cada cada. Cada inteiro v é o valor de uma célula do mapa.

- $1 <= N, M <= 50$
- $-10^9 <= v_{ij} <= 10^9$

Output

Imprima 3 linhas

A primeira linha contendo um inteiro representando a soma do retângulo.

A segunda linha contendo dois inteiros representando as coordenadas do primeiro ponto.

A terceira linha contendo dois inteiros representando as coordenadas do segundo ponto.

Caso houver diversas respostar possíveis, imprima a resposta lexicograficamente menor.

Examples

standard input	standard output
3 2 -3 2 3 -3 -3 2	3 2 1 2 1
2 3 -3 2 -3 4 -3 4	5 2 1 2 3
2 5 -2 1 0 -2 2 -9 -1 2 0 0	2 1 2 2 3
1 1 -100	-100 1 1 1 1

Note

Dizemos que uma resposta s é lexicograficamente menor que t se o primeiro inteiro de s que difere do correspondente em t é menor que seu correspondente.

Problem H. Hora de Brincar com Conjuntos

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

No seu aniversário, Machado ganhou diversos pares de conjuntos de Joãozinho. Confuso pelo presente, ele começou a brincar com o conjunto, e chegou à seguinte pergunta: dado um conjunto K, verificar se K é um subconjunto da seguinte operação em alguma permutação P possível:

$$T = (S_{p_1} \cup S_{p_2}) \cap (S_{p_3} \cup S_{p_4}) \cap ... \cap (S_{p_{n-1}} \cup S_{p_n})$$

Sendo que P, nesse caso, é uma permutação dos elementos entre 1 e n sem repetição entre os elementos. Por exemplo, são permutações de tamanho 4: $\{1, 2, 3, 4\}$, $\{1, 3, 4, 2\}$, $\{1, 4, 2, 3\}$.

Aqui, p_i representa alguma posição no conjunto de conjuntos do Machado. Nesse caso, seja S composto pelos conjuntos $\{1, 3, 5\}, \{2, 3\}, \{1, 4, 6, 7\}, \{1, 2, 3\}$, e uma permutação $P = \{1, 3, 4, 2\}$, p_1 representa o primeiro valor da permutação P, e S_{p_1} representa o conjunto $S_1 = \{1, 3, 5\}$. Da mesma forma, p_2 representa o segundo valor da permutação P e, portanto, S_{p_2} representa o conjunto $S_3 = \{1, 4, 6, 7\}$.

Lembrando que um conjunto não possui repetições entre os seus elementos. O conjunto T é um subconjunto de S, se todo elemento presente em T está presente em S. A união ($\cup$) de dois conjuntos S e T é o menor conjunto U possível onde S e T são subconjuntos de U. A interseção ($\cap$) de dois conjuntos S e T é o maior conjunto U possível que seja ao mesmo tempo subconjunto de S e subconjunto de T.

Input

Na primeira linha, serão fornecidos dois inteiros, N e M, representando o número de conjuntos que o Machado ganhou e o tamanho do conjunto K. É garantido que N é par. Na segunda linha, serão fornecidos M inteiros, representando os elementos k_j do conjunto K. Nas próximas $N * 2$ linhas, serão fornecidas pares de linhas da seguinte forma: Na linha i, será fornecido um inteiro Z, representando o tamanho do conjunto S_i. Na linha $i + 1$, serão fornecidos Z inteiros, representando os elementos s_{i_j} do conjunto S_i.

Limites:

- $1 \leq N \leq 2000$
- $1 \leq M \leq 13$
- $1 \leq Z \leq 100$
- $1 \leq k_j, s_{i_j} \leq 10^5$

Output

Imprima "SIM", sem aspas, caso exista uma permutação onde K é um subconjunto de T e "NAO", sem aspas, caso contrário.

Example

standard input	standard output
4 3 1 2 3 2 1 2 2 3 4 2 2 3 4 1 2 3 4	SIM

Note

Explicação do exemplo:

Aqui, temos os conjuntos $K = \{1, 2, 3\}$, $S_1 = \{1, 2\}$, $S_2 = \{3, 4\}$, $S_3 = \{2, 3\}$ e $S_4 = \{1, 2, 3, 4\}$. Algumas permutações possíveis são $P = \{1, 2, 3, 4\}$ e $P = \{1, 3, 2, 4\}$. Uma permutação que não gera K como subconjunto seria $P = \{1, 4, 2, 3\}$, que teria como resultado da operação $T = \{2, 3\}$ e, portanto, K não seria um subconjunto de T.

Problem I. Isso Reduz Para O Que?

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

Certo dia, Joãozinho estava estudando sobre uma forma de reduzir o problema das N-Rainhas para um problema SAT. Em um dado momento, quando Joãozinho já estava com sono, alguém perguntou se o SAT gerado era um 2-SAT, e a resposta foi que não, pois existia uma cláusula de tamanho 4. Joãozinho, porém, quase respondeu que, se a cláusula fosse de tamanho 3, era possível através de uma redução usando 3 cláusulas de tamanho 2. Mas, logicamente, a ideia de Joãozinho estava errada, pois caso contrário ele estaria escrevendo um artigo para ficar milionário, e não escrevendo probleminhas para a maratona.

Joãozinho lembrou que, dado uma fórmula SAT qualquer, a mesma pode sempre ser reduzida para uma fórmula 3-SAT-FNC. Porém, se a fórmula for 1-SAT ou 2-SAT, não vale a pena reduzir a mesma para 3-SAT-FNC, bem como não vale a pena reduzir uma fórmula 1-SAT para uma fórmula 2-SAT. Além disso, não se sabe se pode reduzir uma fórmula 3-SAT para 2-SAT, e não se pode reduzir uma fórmula 2-SAT para 1-SAT.

Uma fórmula SAT na forma normal conjuntiva (FNC) consiste de uma conjunção de cláusulas de disjunção de literais, que são proposições atômicas (i.e. a forma mais simples de uma fórmula proposicional, que pode ser valorada como positiva ou negativa). Os literais são representados através de letras e números como, por exemplo: $p, q, r, p_1, p_2, p_3, ...$, podendo ser um literal positivo p ou um literal negativo $\sim p$. As disjunções são representadas pelo símbolo $\vee$, e as cláusulas são delimitadas por parênteses (). As conjunções são representadas pelo símbolo $\wedge$. Sendo assim, $(p \vee q) \wedge (r)$ está na FNC, mas as fórmulas $(p \wedge q) \vee (r)$ e $(p \vee q) \wedge r$ não estão na FNC.

Dado uma fórmula SAT na FNC, dizemos que ela é do tipo 3-SAT-FNC se todas as cláusulas da fórmula possuem **exatamente** 3 literais. Além disso, a fórmula é do tipo 2-SAT se todas as cláusulas da fórmula possuem **no máximo** 2 literais, e são do tipo 1-SAT se todas as cláusulas da fórmula possuem **exatamente** 1 literal.

Por isso, Joãozinho pensou no seguinte problema de maratona: dado uma fórmula SAT na FNC, dizer qual o menor de tamanho de fórmula SAT que essa fórmula pode ser reduzida, ou seja, se ela pode ser reduzida para 1-SAT, 2-SAT ou 3-SAT.

Input

A entrada consiste de uma string contendo letras, números parênteses, o símbolo $\sim$, o símbolo & e o símbolo |, sem espaços em branco.

O símbolo $\sim$ será usado para representar negações, o símbolo & será usado para representar conjunções e o símbolo | será usado para representar as disjunções. Se for copiar algum dos símbolos, é recomendado copiar dos casos de exemplo.

É garantido que a string representa uma fórmula na FNC, que a fórmula FNC não será vazia e que a fórmula FNC não deverá ser simplificada antes de se aplicar a redução, caso a redução seja aplicável.

Limites:

- $0 \le |s| \le 10^5$

Output

Você deve imprimir uma string com a resposta esperada para o problema, podendo ser:

- 1-SAT
- 2-SAT

- 3-SAT

Examples

standard input	standard output
`(a|b|~a)&(c)&(a|b)`	`3-SAT`
`(a|b)&(c|d)`	`2-SAT`
`(a)&(b)&(c)`	`1-SAT`

Problem J. Jogo do Omar

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

É comum amigos se juntarem para jogar algum jogo de baralho, e algumas pessoas são muito competitivas e levam esses jogos muito a sério. Omar é uma dessas pessoas. Atualmente ele está treinando para jogar um famoso jogo de carta onde é necessário formar trincas e sequencias.

Lembrando que uma trinca são três cartas de naipes diferentes do mesmo valor e uma sequência são três cartas do mesmo naipe em ordem.

Omar está jogando com um baralho de N cartas, cada carta pode ser representada como duas letras, v representando o seu valor e t representando o seu naipe.

O treinamento de Omar consiste em começar com a mão vazia e, a cada rodada, comprar uma carta do baralho. Se estiver com mais de nove cartas ele descarta a carta que está a mais rodadas na sua mão. Infelizmente Omar não consegue contar cartas e contar rodadas ao mesmo tempo, então ele precisa da sua ajuda.

Sabendo a ordem das cartas no baralho de Omar, quantas rodadas são necessárias até que ele forme a primeira trinca em sua mão?

Input

A primeira linha da entrada contem um inteiro N.

Em seguida seguem N linhas, cada linha contendo dois caracteres, v e t.

- $0 \leq N \leq 10^5$
- $v \in \{A, 2, 3, 4, 5, 6, 7, 8, 9, 0, J, Q, K\}$
- $t \in \{O, C, P, E\}$

Output

Imprima um unico inteiro representando quantas rodadas são necessárias até que Omar forme a primeira trinca em sua mão. Se o baralho acabar e Omar não formar nenhuma trinca, imprima -1.

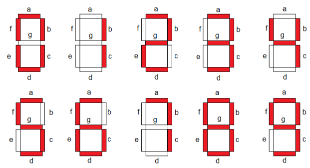
Examples

standard input	standard output
5 KO JO KC KP KE	4
10 KO JO KC 2P JE AO AC OP OE KP	-1

Problem K. Kazuba

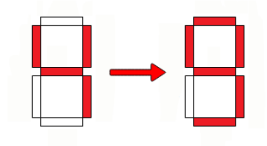
Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Enquanto corrigia a provas de MDI (Mostradores Digitais) o professor Kazuba quis trazer um pouco de entretenimento para esse processo tão chato de correção. Ele decidiu que daria as notas seguindo o famoso Seven-segment display (veja figura abaixo), ele imprimiu as marcas do display e só precisava pintar os segmentos com caneta para indicar as notas.



Seven-segment display

Após entregar as provas para que os alunos dessem uma olhadinha, ele pediu para que elas fossem devolvidas a fim de guardá-las e mantê-las devidamente documentadas. Quando estava recolhendo as provas, o professor notou que alguns alunos haviam adulterado as notas adicionado segmentos de forma a representar outro valor válido.



Possível troca que os alunos fizeram

Para evitar este problema, o professor começou um novo projeto de pesquisa. Ele quer desenvolver um conjunto de símbolos para displays de M segmentos (observe que no Seven-Segment display M=7), no qual todos os símbolos válidos não podem ser adulterados pela adição de novos segmentos e resultarem em outro símbolo válido. Como o professor não gosta de manter suas notas sempre entre 0 e 10, ele quer desenvolver displays que possam mostrar ao menos N notas distintas (cada nota deve ser representada por uma combinação distinta de segmentos pintados).

A sua tarefa é dado uma quantidade de notas distintas N, dizer o número mínimo de segmentos, M, necessários no display de tal forma que possam existir ao menos N notas distintas que respeitem os pedidos do professor.

Input

A primeira linha da entrada consiste de um inteiro T indicando a quantidade de casos de testes.

Para cada caso de teste será dado um inteiro N que indica a quantidade de notas distintas que devem ser representadas.

- $1 \le T \le 10^5$.
- $1 \le N \le 10^{15}$.

Output

Para cada caso de teste você deve imprimir o número mínimo de segmentos, M, que devem haver no display para representar ao menos N notas distintas.

Examples

standard input	standard output
2 5 19	4 6
3 1 2 43	0 2 8

Problem L. Lucrando com Encanamentos

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

Granza é o dono de uma empresa de conserto de encanamentos em Nlogônia chamada BRUTE Plumbing Repair. Recentemente, ele recebeu um chamado para consertar o encanamento de toda Nlogônia, onde diversos segmentos do encanamento estão quebrados. Sabendo da capacidade de seus funcionários consertarem quantos metros forem possíveis num mesmo dia, ele aceitou o prazo que foi passado, mesmo que esse prazo exija muito dos reparadores da BRUTE, porque uma grande quantidade de dinheiro estava em jogo, e ele poderia usar esse dinheiro para comprar o que ele mais deseja no momento, uma Harley-Davidson.

O encanamento de Nlogônia consiste de uma linha reta de vários metros, e os segmentos foram passados para Granza como intervalos sequenciais, ou seja, com os pontos iniciais ordenados, sem nenhuma interseção entre os segmentos. Granza sabe também o quão exigente são seus funcionários no reparo, de tal forma que eles não gostam de dividir o conserto de um mesmo segmento com outro funcionário, e não gostam quando a ordem dos segmentos que eles recebem não são contíguos. Por exemplo, se for necessário reparar os segmentos $1 : [1, 2]$, $2 : [5, 7]$ e $3 : [9, 10]$, um encanador não gostaria de receber os segmentos 1 e 3, mas não se importaria de receber qualquer outra ordem, nesse caso, como 1 e 2, 2 e 3 ou 1, 2 e 3.

Como os seus funcionários gostam de receber igualmente, todos recebem a mesma quantidade de dinheiro que o funcionário que mais trabalhou no dia, e todos acham esse modelo aceitável, pois eles revezam entre uma obra e outra quem vai trabalhar mais. Sendo assim, para estimar o preço da obra e ver se de fato será possível comprar uma Harley-Davidson após pagar os funcionários, Granza pediu sua ajuda.

A estimativa será feita da seguinte forma. Será estimado uma metragem por dia, e a estimativa de pagamento será o preço pago por metro reparado multiplicado pela metragem estipulada e pelo número de dias da obra. É considerado que cada encanador irá reparar **no máximo** a metragem estimada a cada dia.

Realizar o cálculo da estimativa é fácil mas, para saber qual o menor preço estimado possível, você deve calcular qual deve ser a menor metragem consertada por dia, de tal forma que o encanamento de Nlogônia seja reparado no prazo que foi passado para Granza, e respeitando as preferências dos encanadores.

Input

Na primeira linha serão fornecidos 4 inteiros N, M, C e D, que representam, respectivamente, o número de metros do encanamento de Nlogônia, a quantidade de segmentos, a quantidade de encanadores disponíveis e o prazo passado para Granza.

M linhas seguem, cada uma com dois inteiros x e y, representando, respectivamente, o ponto de início e o ponto final de cada segmento. É garantido que nenhum segmento se interseciona e que os segmentos são dados em ordem.

Limites:

- $2 \le N \le 10^{18}$
- $1 \le M, C, D \le 2 * 10^5$
- $1 \le x < y \le N$

Output

Você deve imprimir um único inteiro M_d, representando a estimativa da menor metragem a ser reparada por dia, respeteitando as condições do problema.

Examples

standard input	standard output
10 2 2 3 1 5 6 10	2
15 4 1 3 1 4 5 6 9 12 14 15	3
500 4 2 4 1 3 100 112 200 202 300 307	4

Note

No terceiro caso de teste, um encanador consertará os 2 metros do primeiro segmento e outros 2 metros do segundo segmento no primeiro dia, e nos outros 3 dias ficará consertando o segundo segmento. O segundo encanador consertará os dois metros do terceiro segmento e outros 2 metros do quarto segmento, e terminará de consertar o quarto segmento no terceiro dia, descansando assim no quarto dia. Note que, caso não existisse a restrição dos segmentos contíguos, a resposta seria 3, com um encanador reparando os segmentos 1, 3 e 4, e outro encanador reparando o segundo segmento.

Problem M. Marcha

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

Marquiz é um exímio competidor de marcha atlética. Na flor da idade, está aproveitando que tem algum tempo livre para se preparar fisicamente para suas próximas competições. Apesar de parecer um esporte simples, a marcha atlética envolve muita técnica e ritmo. Quanto ao ritmo, Marquiz não tem problema algum, mas está com dificuldades de aprimorar sua técnica pois o terreno onde treina possui alguns desagradáveis desníveis.

Marquiz então entrou em contato com seus colegas da prefeitura e conseguiu um mapa dos vários bairros da cidade com os seus respectivos níveis de cada região. Sua tarefa agora é ajudá-lo a definir com base no mapa fornecido qual a menor quantidade de mudança de nível Marquiz terá que enfrentar em cada um dos bairros fornecidos.

Input

Os mapas que Marquiz conseguiu estão no formato de uma matriz M que possui ordem LxC. Sua casa está localizada na coordenada $M_{0,0}$ e a academia (seu destino) se encontra na coordenada $M_{L-1,C-1}$. Marquiz é bem metódico com seu exercício, ele só caminha nos sentidos ortogonais (norte, sul, leste e oeste) e apenas da pequenos passos de tamanho 1 (i.e., só é possível se movimentar de (x_1, y_1) para (x_2, y_2) se e somente se $(|x_2 - x_1| + |y_2 - y_1|) == 1$, $0 \leq x_2 < L$ e $0 \leq y_2 < C$). Cada diferente nível de rua será diferenciado por uma letra entre A e Z.

A entrada inicia com um inteiro T indicando a quantidade de bairros que deverão ser avaliados. Para cada bairro é esperado uma análise.
A primeira linha dos T casos de teste consiste de 2 inteiros L, C. As próximas L linhas conterão C letras entre $'A'$ e $'Z'$.

- $1 \leq T \leq 10$.
- $2 \leq R, C \leq 1000$.
- $'A' \leq M_{i,j} \leq 'Z'$.

Output

Você deve imprimir, para cada um dos T bairros, a quantidade mínima de desníveis que Marquiz enfrentará.

Examples

standard input	standard output
2 2 2 AA AA 2 3 ABC DEF	0 3
2 6 6 AKACCC AAACFC AMDFCC AOKHDD ZYXWDP ZYXWDD 5 5 ABBBC ABACC AAACC AEFCI CDGDD	2 2

Problem N. Navio

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Uma forma popular de transportar mercadorias é utilizando contêineres.

Bob gerencia o carregamento de contêineres para transporte por navio, sendo responsável por determinar a forma ótima de preencher contêineres de modo a maximizar seus lucros. Existem N contêineres que Bob pode carregar, sendo que o i-ésimo contêiner só pode ser carregado com mercadorias de rótulo i. Sabendo o peso máximo P do navio, e o peso máximo individual a_i de cada contêiner, Bob precisa determinar quais contêineres serão carregados no navio, e quais mercadorias serão carregadas em cada contêiner.

Cada mercadoria j possui um peso b_j e um lucro x_j. Contêineres podem ser carregados com qualquer peso de mercadorias, contanto que não passe do limite máximo individual daquele contêiner. O navio pode ser carregado com qualquer peso, contanto que não extrapole seu peso limite. Por questões de segurança, é sempre considerado o peso máximo do contêiner quando ele é colocado no navio. Ou seja, a carga total do navio é a soma dos pesos máximos de cada contêiner e não a soma de seus pesos reais, considerando as mercadorias carregadas.

Bob sempre realizou essa otimização manualmente, utilizando duas folhas de papel, uma caneta azul, e uma planilha no Excel. Porém, devido ao crescimento de seu negócio, esse processo deixou de ser viável. Para escalar sua operação, Bob deseja automatizar a otimização do carregamento de contêineres, e para isso contratou os serviços de um expert. Infelizmente esse profissional acabou adoecendo na última hora, e portanto Bob decidiu pedir sua ajuda para resolver esse problema.

Input

A primeira linha do input consiste em três inteiros: o número de contêineres $1 \leq N \leq 5000$, o número de mercadorias $1 \leq M \leq 5000$, e o peso máximo do návio $1 \leq P \leq 5000$.

A segunda linha consiste em N inteiros $1 \leq a_i \leq 5000$, onde a_i é o peso máximo do návio i.

As próximas M linhas consistem em três inteiros: o peso $1 \leq b_j \leq 5000$, o lucro $1 \leq x_j \leq 10^9$, e o rótulo $1 \leq k_j \leq N$ da mercadoria j.

É garantido que a soma de todos os a_i não é superior a 10^4.

Output

Imprima o lucro máximo que pode ser obtido se o navio for carregado de maneira ótima.

Examples

standard input	standard output
5 4 10 1 2 3 4 5 1 8 1 2 5 2 2 3 3 3 2 5	16
3 6 20 5 10 20 3 2 1 10 10 1 7 5 2 5 8 2 6 3 3 14 1 3	10