



Maratona de Programação da SBC 2015

Sub-Regional Brasil do ACM ICPC

12 de Setembro de 2015

Caderno de Problemas

Informações Gerais

Este caderno contém 12 problemas; as páginas estão numeradas de 1 a 15, não contando esta página de rosto. Verifique se o caderno está completo.

A) Sobre os nomes dos programas

1) Sua solução deve ser chamada `codigo_de_problema.c`, `codigo_de_problema.cpp` ou `codigo_de_problema.java`, onde `codigo_de_problema` é a letra maiúscula que identifica o problema. Lembre que em Java o nome da classe principal deve ser igual ao nome do arquivo.

B) Sobre a entrada

- 1) A entrada de seu programa deve ser lida da *entrada padrão*.
- 2) A entrada é composta de um único caso de teste, descrito em um número de linhas que depende do problema.
- 3) Quando uma linha da entrada contém vários valores, estes são separados por um único espaço em branco; a entrada não contém nenhum outro espaço em branco.
- 4) Cada linha, incluindo a última, contém exatamente um caractere final-de-linha.
- 5) O final da entrada coincide com o final do arquivo.

C) Sobre a saída

- 1) A saída de seu programa deve ser escrita na *saída padrão*.
- 2) Quando uma linha da saída contém vários valores, estes devem ser separados por um único espaço em branco; a saída não deve conter nenhum outro espaço em branco.
- 3) Cada linha, incluindo a última, deve conter exatamente um caractere final-de-linha.

Promoção:



Sociedade Brasileira de Computação

Problema A

Mania de Par

Patrícia é uma ótima desenvolvedora de software. No entanto, como quase toda pessoa brilhante, ela tem algumas manias estranhas, e uma delas é que tudo que ela faz tem que ser em número par. Muitas vezes essa mania não atrapalha, apesar de causar estranhamento nos outros. Alguns exemplos: ela tem que fazer diariamente um número par de refeições; no café da manhã toma duas xícaras de café, duas torradas e duas fatias de queijo; sempre que vai ao cinema compra dois bilhetes de entrada (felizmente sempre tem um amigo ou amiga lhe acompanhando); e toma dois banhos por dia (ou quatro, ou seis...).

Mas algumas vezes essa mania de Patrícia atrapalha. Por exemplo, ninguém gosta de viajar de carro com ela, pois se no trajeto ela tem que pagar pedágios, o número de pedágios que ela paga tem que ser par.

Patrícia mora em um país em que todas as estradas são bidirecionais e têm exatamente um pedágio. Ela precisa ir visitar um cliente em uma outra cidade, e deseja calcular o mínimo valor total de pedágios que ela tem que pagar, para ir da sua cidade à cidade do cliente, obedecendo à sua estranha mania de que o número de pedágios pagos tem que ser par.

Entrada

A entrada consiste de diversas linhas. A primeira linha contém 2 inteiros C e V , o número total de cidades e o número de estradas ($2 \leq C \leq 10^4$ e $0 \leq V \leq 50000$). As cidades são identificadas por inteiros de 1 a C . Cada estrada liga duas cidades distintas, e há no máximo uma estrada entre cada par de cidades. Cada uma das V linhas seguintes contém três inteiros C_1 , C_2 e G , indicando que o valor do pedágio da estrada que liga as cidades C_1 e C_2 é G ($1 \leq C_1, C_2 \leq C$ e $1 \leq G \leq 10^4$). Patrícia está atualmente na cidade 1 e a cidade do cliente é C .

Saída

Uma única linha deve ser impressa, contendo um único inteiro, o custo total de pedágios para Patrícia ir da cidade 1 à cidade C , pagando um número par de pedágios, ou, se isso não for possível, o valor -1 .

Exemplos

Entrada	Saída
4 4 1 2 2 2 3 1 2 4 10 3 4 6	12

Entrada	Saída
5 6 1 2 3 2 3 5 3 5 2 5 1 8 2 4 1 4 5 4	-1

Problema B

Bolsa de Valores

Um investidor principiante deseja aprender a investir na bolsa de valores. Como ele não tem experiência, selecionou uma única empresa, e acompanhou os valores diários das ações dessa empresa, durante N dias. Ficou curioso quanto teria ganhado se tivesse investido nesse período em que acompanhou os valores. Na verdade, o investidor é milionário e tem muito dinheiro, suficiente para comprar qualquer quantidade de ações da empresa. Entretanto, como é um investidor cuidadoso, decidiu que nunca teria mais do que uma ação da empresa.

Como sempre há intermediários, a corretora de valores cobra uma taxa fixa de C reais a cada compra de uma ação da empresa.

Você deve calcular qual o lucro máximo que o investidor poderia ter auferido, investindo durante alguns dos N dias, podendo inclusive decidir não investir.

Entrada

A primeira linha contém dois inteiros, N e C ($1 \leq N \leq 2 \times 10^5$ e $0 \leq C \leq 30$).

A segunda linha contém as N cotações P_1, P_2, \dots, P_N , dos dias $1, 2, \dots, N$, respectivamente. Cada cotação P_i satisfaz as desigualdades $1 \leq P_i \leq 1000$.

Saída

Seu programa deve produzir uma única linha com um inteiro representando o lucro máximo do investidor, em reais.

Exemplos

Entrada 6 10 100 120 130 80 50 40	Saída 20
Entrada 5 10 70 80 50 40 50	Saída 0
Entrada 13 30 10 80 20 40 30 50 40 60 50 70 60 10 200	Saída 220

Problema C

Tri-du

Tri-du é um jogo de cartas derivado do popular jogo de Truco. O jogo utiliza um baralho normal de 52 cartas, com treze cartas de cada naipe, mas os naipes são ignorados. Apenas o valor das cartas, considerados como inteiros de 1 a 13, são utilizados.

No jogo, cada jogador recebe três cartas. As regras são simples:

- Um trio (três cartas de mesmo valor) ganha de uma dupla (duas cartas de mesmo valor).
- Um trio formado por cartas de maior valor ganha de um trio formado por cartas de menor valor.
- Uma dupla formada por cartas de maior valor ganha de uma dupla formada por cartas de menor valor.

Note que o jogo pode não ter ganhador em muitas situações; nesses casos, as cartas distribuídas são devolvidas ao baralho, que é embaralhado e uma nova partida é iniciada.

Um jogador já recebeu duas das cartas que deve receber, e conhece seus valores. Sua tarefa é escrever um programa para determinar qual o valor da terceira carta que maximiza a probabilidade de esse jogador ganhar o jogo.

Entrada

A entrada consiste de uma única linha que contém dois inteiros, A ($1 \leq A \leq 13$) e B ($1 \leq B \leq 13$) indicando os valores das duas primeiras cartas recebidas.

Saída

Seu programa deve produzir uma única linha com um inteiro representando o valor da carta que maximiza a probabilidade de o jogador ganhar a partida.

Exemplos

Entrada 10 7	Saída 10
Entrada 2 2	Saída 2

Problema D

Quebra-cabeça

Discussões recentes na Internet causaram uma onda de renovado interesse em quebra-cabeças de lógica. Neste problema a sua tarefa é escrever um programa que resolva quebra-cabeças como o mostrado na figura abaixo, muito comum em revistas de desafios lógicos. Nesse quebra-cabeças, as letras dentro do quadriculado representam variáveis, e os números representam as somas dos valores das variáveis em cada linha ou coluna.

df	bb	cg	df	df	⇒	11
ee	az	cg	az	ee	⇒	6
df	cg	cg	df	df	⇒	10
az	az	cg	az	az	⇒	6
↓	↓	↓	↓	↓		
6	7	8	6	6		

O objetivo desse tipo de quebra-cabeça é determinar o valor de cada variável de modo a satisfazer as somas das linhas e colunas mostradas. Mas como esse tipo de quebra-cabeças é para crianças, ele tem uma propriedade que o torna mais fácil de encontrar a solução: sempre é possível encontrar uma linha ou coluna em que há apenas uma variável cujo valor ainda é desconhecido. Assim, uma possível maneira de resolver o problema é, a cada passo da solução, encontrar o valor de uma variável.

Dado um quebra-cabeça, você deve determinar os valores das variáveis que o solucionam.

Entrada

A primeira linha contém dois inteiros L ($1 \leq L \leq 100$) e C ($1 \leq C \leq 100$) indicando o número de linhas e o número de colunas do quebra-cabeça. Cada uma das L linhas seguintes contém C nomes de variáveis, seguidos de um inteiro S , a soma resultante das variáveis dessa linha ($-10^8 \leq S \leq 10^8$). A última linha contém C inteiros X_i ($-10^8 \leq X_i \leq 10^8$), indicando respectivamente a soma das variáveis na coluna i . Nomes de variáveis são formados por precisamente duas letras minúsculas, de 'a' a 'z'. Todos os quebra-cabeças têm solução única, em que todas as variáveis são números inteiros.

Saída

Seu programa deve produzir uma linha para cada variável do quebra-cabeças, contendo o nome da variável e o seu valor inteiro. As variáveis devem ser escritas em ordem alfabética crescente, ou seja, respeitando a ordem

aa, ab, ..., az, ba, bb, ..., za, zb, ..., zz.

Exemplos

Entrada	Saída
4 5	az 1
df bb cg df df 11	bb 3
ee az cg az ee 6	cg 2
df cg cg df df 10	df 2
az az cg az az 6	ee 1
6 7 8 6 6	

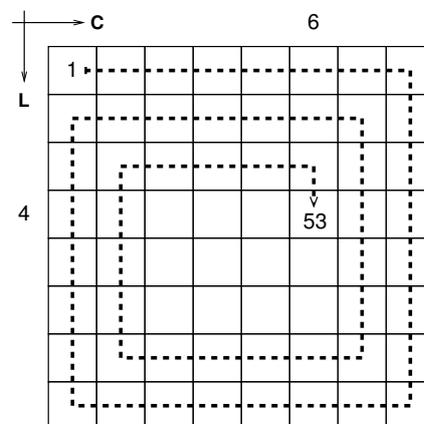
Entrada	Saída
3 4 aa bb cc dd 10 aa bb cc dd 10 aa bb cc dd 10 3 6 9 12	aa 1 bb 2 cc 3 dd 4

Entrada	Saída
3 3 aa zz aa 27 vv zz aa -5 kk kk aa 40 15 -7 54	aa 18 kk 11 vv -14 zz -9

Problema E

Espiral

Dado um tabuleiro de dimensões $N \times N$, gostaríamos de colocar feijões, um grão em cada quadrado, seguindo uma espiral como mostrado na figura. Começando do canto superior esquerdo, com coordenadas $(1, 1)$, e depois indo para a direita enquanto possível, depois para baixo enquanto possível, depois para esquerda enquanto possível e depois para cima enquanto possível. Repetimos esse padrão, direita-baixo-esquerda-cima, até que B grãos de feijão sejam colocados no tabuleiro. O problema é: dados N e B , em que coordenadas será colocado o último grão de feijão? Na figura, para $N = 8$ e $B = 53$, o último grão foi colocado no quadrado de coordenadas $(4, 6)$.



Entrada

A entrada contém apenas uma linha com dois inteiros, N e B , onde $1 \leq N \leq 2^{30}$ and $1 \leq B \leq N^2$.

Saída

Seu programa deve produzir uma única linha com dois inteiros L e C representando as coordenadas do último grão de feijão.

Exemplos

Entrada 8 53	Saída 4 6
Entrada 1073741824 1152921504603393520	Saída 536871276 536869983

Problema F

Fatorial

O *fatorial* de um número inteiro positivo N , denotado por $N!$, é definido como o produto dos inteiros positivos menores do que ou iguais a N . Por exemplo $4! = 4 \times 3 \times 2 \times 1 = 24$.

Dado um inteiro positivo N , você deve escrever um programa para determinar o menor número k tal que $N = a_1! + a_2! + \dots + a_k!$, onde cada a_i , para $1 \leq i \leq k$, é um número inteiro positivo.

Por exemplo, para $N = 10$ a resposta é 3, pois é possível escrever N como a soma de três números fatoriais: $10 = 3! + 2! + 2!$. Para $N = 25$ a resposta é 2, pois é possível escrever N como a soma de dois números fatoriais: $25 = 4! + 1!$.

Entrada

A entrada consiste de uma única linha que contém um inteiro N ($1 \leq N \leq 10^5$).

Saída

Seu programa deve produzir uma única linha com um inteiro representando a menor quantidade de números fatoriais cuja soma é igual ao valor de N .

Exemplos

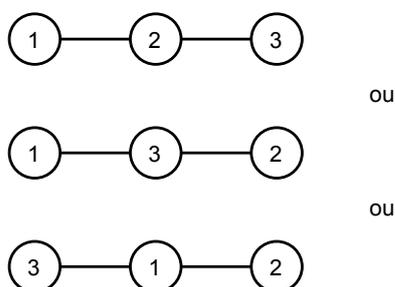
Entrada	Saída
10	3
25	2

Problema G

Guardiões Curiosos

Oa é um dos mundos mais antigos do universo DC, é lá que habitam os guardiões do universo. Eles administram a tropa dos lanternas verdes, uma das maiores forças do universo! Todos sabem que os lanternas verdes sabem voar devido ao poder do anel, porém nem todos os habitantes de Oa fazem parte da tropa. Para esses habitantes está difícil se locomover entre as cidades, pois não há estradas!

Os guardiões desejam conectar as cidades de Oa construindo algumas estradas. Existem N cidades em Oa, e eles desejam construir $N - 1$ estradas de duas mãos, de tal forma que seja possível chegar de uma cidade até qualquer outra, direta ou indiretamente. Os guardiões também não desejam privilegiar demais nenhuma cidade, por isso eles estabeleceram que nenhuma cidade pode ter mais de K estradas. Por exemplo, se temos três cidades e K vale 2, temos as três opções:



Os guardiões, porém, são muito curiosos, e perguntaram aos lanternas verdes se eles eram capazes de dizer de quantas formas é possível construir $N - 1$ estradas obedecendo estas restrições. Sua tarefa, como membro da tropa dos lanternas verdes é, dados N e K , satisfazer a curiosidade dos guardiões.

Entrada

A entrada consiste de uma única linha que contém dois números inteiros N ($1 \leq N \leq 10^2$) e K ($1 \leq K \leq N$).

Saída

Seu programa deve produzir uma única linha, contendo um único número inteiro, a resposta do problema. Como essa resposta pode ser muito grande, imprima-a módulo $10^9 + 7$.

Exemplos

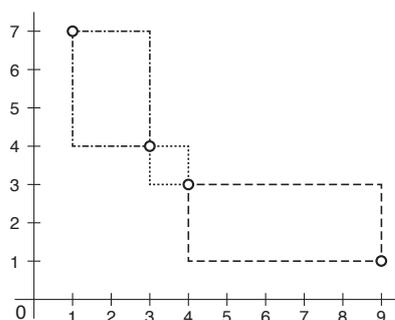
Entrada 3 2	Saída 3
Entrada 4 1	Saída 0
Entrada 4 3	Saída 16

Problema H

Praça do Retângulo

Retangolândia é uma cidade muito antiga e, por isso, guarda diversas riquezas históricas. A cidade foi planejada muitas décadas atrás, com todas as suas ruas indo nas direções norte-sul ou leste-oeste. Atualmente, há um projeto de revitalização da cidade, no qual uma nova praça retangular será feita. A escolha da nova praça será feita pela administração pública mas, no momento, eles estão interessados em quais seriam as posições possíveis para esta praça, levando-se em consideração que a praça deve estar alinhada com as ruas e, assim, quando visualizada em um mapa, seus lados devem ser segmentos horizontais e verticais. Com o objetivo de conciliar as riquezas históricas com as novas iniciativas, alguns cuidados devem ser tomados.

Existem postes de iluminação, do século XIX, espalhados pela cidade. Por seu valor histórico, nenhum poste pode ser derrubado. Por conta do desgaste natural e da falta de manutenção, nenhuma rua possui mais do que um poste restante. Para o posicionamento da praça, entretanto, não se deseja que um destes postes esteja no interior da mesma. Por outro lado, o projeto paisagístico da nova praça prevê que dois dos postes históricos estejam em duas das esquinas. A figura abaixo mostra um exemplo com quatro postes e as três localizações possíveis para a praça.



A prefeitura contratou uma empresa de georeferenciamento para efetuar um levantamento das posições dos postes. Com esses dados em mãos, o próximo passo é determinar quantas são as localizações possíveis para a praça, para que se possa dimensionar o tamanho da equipe necessária para avaliar cada uma das localizações.

Entrada

A primeira linha da entrada contém um número inteiro N , $1 \leq N \leq 3000$, representando o número de postes. As N linhas seguintes descreverão, cada uma, a posição de um poste. A posição de um poste será dada por um par de números inteiros, X e Y , $-10^8 \leq X, Y \leq 10^8$, correspondendo às suas coordenadas no plano.

Saída

Seu programa deve produzir uma única linha contendo o número de diferentes localizações possíveis para a praça.

Exemplos

Entrada	Saída
4 1 7 4 3 3 4 9 1	3

Entrada	Saída
5 1 7 5 5 2 2 8 8 6 -1	8

Entrada	Saída
8 1 1 2 2 -2 200 100 3 -6 -6 -51 19 -3 -1 8 -2	19

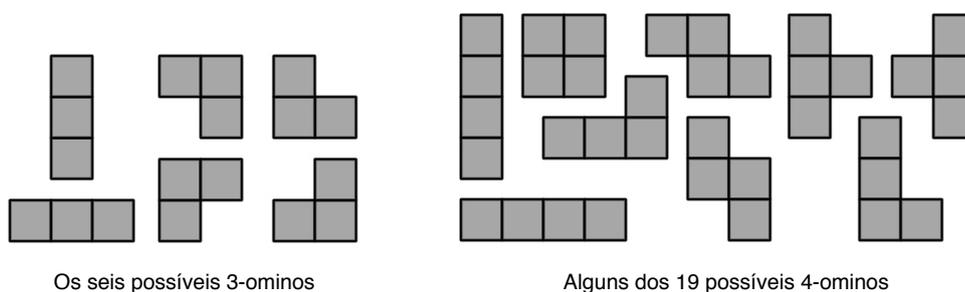
Problema I

Ominobox

O planeta de S kyrk nunca vai conhecer a paz enquanto o malvado Mago estiver livre. Dessa vez, o malicioso plano do Mago foi armar uma bomba no meio da maior cidade do planeta. Mago aprecia observar o caos, então, ao invés de explodir a bomba imediatamente, ele colocou um temporizador na bomba e a deixou junto com um desafio. A bomba tem um teclado, e a solução do desafio desarma a bomba.

O desafio se chama Ominobox; ele consiste de uma caixa retangular com alguns cubos unitários dentro e de uma coleção de todos os possíveis N -ominos. S kyrk deve soltar todo omino em algum lugar da caixa para ganhar pontos. A pontuação máxima é a solução do Ominobox.

Um N -omino é uma coleção de N quadrados unitários arranjados com lados coincidentes. Um 1-omino é um quadrado unitário, e um N -omino é um $(N - 1)$ -omino com pelo menos um dos seus lados ligados a um quadrado unitário.



Os seis possíveis 3-ominos

Alguns dos 19 possíveis 4-ominos

A caixa tem uma superfície retangular e paredes verticais; cada um dos quadrados de um sistema Cartesiano de coordenadas em grade colocado na superfície da caixa possui uma pilha não negativa de cubos unitários. Os cubos não podem ser movidos.

S kyrk irá alinhar cada omino com os quadrados da grade, e soltá-lo na caixa. O omino irá cair até tocar um cubo ou o fundo. Não é permitido que S kyrk reflita ou rotacione o omino, e ele deve situar-se completamente dentro dos limites da caixa. O número de pontos obtidos após soltá-lo é a distância entre o omino e o topo da caixa. Após soltá-lo, S kyrk anota o número de pontos, remove o omino, e solta o próximo. A pontuação final é a soma de todos os pontos.

O tempo está passando e a contagem regressiva na bomba diz 5:00 (cinco horas!). Você consegue descobrir a pontuação máxima que S kyrk pode obter para desarmar a bomba e salvar o destino do planeta das mãos do vil Mago?

Entrada

A primeira linha contém T ($T \leq 200$) — o número de desafios, após essa linha haverá T desafios. Cada desafio começa com uma linha com quatro inteiros R , C , H e N ($1 \leq R, C, H \leq 30$; $1 \leq N \leq 10$) — as dimensões da superfície da caixa são $R \times C$, a altura é H , e a ordem dos ominos é N . Cada uma das próximas R linhas contém C inteiros $H_{i,j}$ ($0 \leq H_{i,j} \leq H$) — o número de cubos no quadrado (i, j) da grade.

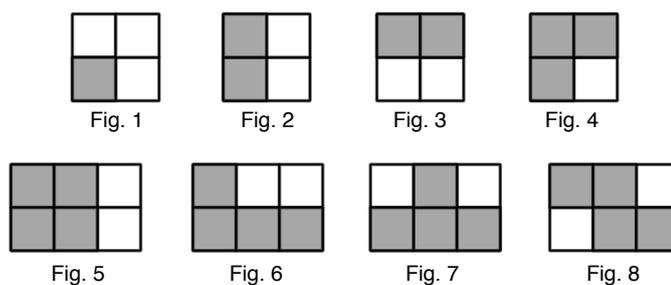
Saída

Para cada desafio, imprima uma linha contendo X , onde X é a solução do Ominobox.

Exemplos

Entrada	Saída
4	3
2 2 3 1	3
1 2	1
0 3	5
2 2 3 2	
1 2	
0 3	
2 2 3 3	
1 2	
0 3	
2 3 5 4	
1 2 5	
0 3 4	

Notas



No primeiro desafio, Fig. 1 mostra a melhor maneira de colocar o único 1-omino. O omino atinge o fundo da caixa na posição (1,0) e possui distância de 3 até o topo da caixa. Esta configuração rende um total de 3 pontos.

No segundo desafio, Fig. 2 e Fig. 3 mostram as melhores maneiras de colocar os dois 2-ominos. Na Fig. 2 o omino atinge a pilha de cubos de altura 1 na posição (0,0) e possui distância de 2 até o topo de caixa. Na Fig. 3 o omino atinge a pilha de cubos de altura 2 na posição (0,1) e possui uma distância de 1 até o topo de caixa. Esta configuração rende um total de 3 pontos.

No terceiro desafio, Fig. 4 mostra a melhor maneira de colocar o único 3-omino que cabe dentro da caixa. Esta configuração rende um total de 1 ponto.

No quarto desafio, Fig. 5-8 mostram as melhores maneiras de colocar os quatro 4-ominos que cabem dentro da caixa. Esta configuração rende um total de 5 pontos.

Problema J

Jogo de Estratégia

Um jogo de estratégia, com J jogadores, é jogado em volta de uma mesa. O primeiro a jogar é o jogador 1, o segundo a jogar é o jogador 2 e assim por diante. Uma vez completada uma rodada, novamente o jogador 1 faz sua jogada e a ordem dos jogadores se repete novamente. A cada jogada, um jogador garante uma certa quantidade de Pontos de Vitória. A pontuação de cada jogador consiste na soma dos Pontos de Vitória de cada uma das suas jogadas.

Dado o número de jogadores, o número de rodadas e uma lista representando os Pontos de Vitória na ordem em que foram obtidos, você deve determinar qual é o jogador vencedor. Caso mais de um jogador obtenha a pontuação máxima, o jogador com pontuação máxima que tiver jogado por último é o vencedor.

Entrada

A entrada consiste de duas linhas. A primeira linha contém dois inteiros J e R , o número de jogadores e de rodadas respectivamente ($1 \leq J, R \leq 500$). A segunda linha contém $J \times R$ inteiros, correspondentes aos Pontos de Vitória em cada uma das jogadas feitas, na ordem em que aconteceram. Os Pontos de Vitória obtidos em cada jogada serão sempre inteiros entre 0 e 100, inclusive.

Saída

Seu programa deve produzir uma única linha, contendo o inteiro correspondente ao jogador vencedor.

Exemplos

Entrada 3 3 1 1 1 1 2 2 2 3 3	Saída 3
Entrada 2 3 0 0 1 0 2 0	Saída 1

Problema K

Palíndromo

Um palíndromo é uma cadeia de caracteres tal que sua reversão é igual à cadeia original. Em outras palavras, é uma cadeia que, quando lida de trás pra frente, é igual à cadeia original. Por exemplo BANANAB é um palíndromo, enquanto BANANAS não. Neste problema estamos interessados em uma questão um pouco mais interessante.

Dada uma cadeia S , queremos encontrar uma subsequência que seja um palíndromo. Uma subsequência é uma cadeia que pode ser obtida a partir da remoção de zero ou mais caracteres da cadeia original. Por exemplo ANNA é uma subsequência de BANANAS.

Será dado também um conjunto de posições de S que chamamos de posições especiais. Sua tarefa é encontrar o tamanho da subsequência que seja um palíndromo e que contenha o maior número de posições especiais possível. Caso exista mais de uma subsequência maximizando o número de posições especiais, você deve imprimir o tamanho da maior delas.

Entrada

A entrada consiste de duas linhas. A primeira linha contém uma cadeia de caracteres maiúsculos S com pelo menos 1 e no máximo 2000 caracteres. A segunda linha contém um inteiro N , $0 \leq N \leq |S|$, indicando o número de posições especiais que estamos interessados em incluir no palíndromo, seguido de N números distintos, entre 1 e $|S|$, inclusive, contendo as posições especiais de S .

Saída

Seu programa deve imprimir um único inteiro, representando o tamanho do maior palíndromo possível, como definido acima.

Exemplos

Entrada BANANAS 0	Saída 5
Entrada BANANAS 1 7	Saída 1
Entrada ACDAAACX 3 2 3 8	Saída 3
Entrada MARATONA 4 3 1 5 2	Saída 3

Problema L

Loteria

A loteria BWS é feita anualmente. Nela N pessoas apostam escolhendo K números cada uma. De modo formal, podemos dizer que B_{ij} é o j -ésimo valor apostado pela i -ésima pessoa. Então os organizadores escolhem K inteiros positivos. Os números escolhidos são chamados de W_1, W_2, \dots, W_K .

Os vencedores são calculados da seguinte maneira:

- Um subconjunto não vazio dos N participantes é escolhido aleatoriamente, ou seja, alguns participantes são escolhidos por pura sorte.
- Para cada pessoa neste subconjunto é calculado o valor S_i , que é a soma de todos os primeiros números apostados por elas, ou seja, a soma de B_{i1} , onde i seria o índice de cada pessoa escolhida. Da mesma maneira os valores S_2, \dots, S_K são calculados.
- É feito um teste de paridade entre W_j e S_j , ou seja, é testado se as paridades (se o número é par ou ímpar) casam entre W_1 e S_1 , W_2 e S_2 , e assim por diante até W_K e S_K .
- Se todas as paridades casam, então este conjunto de pessoas é considerado vencedor!

Os organizadores querem saber: é possível escolher os números W_1, W_2, \dots, W_K de forma que **não** exista nenhum subconjunto de participantes vencedor?

Entrada

A primeira linha contém os números N ($1 \leq N \leq 10^4$) e K ($3 \leq K \leq 50$), representando o número de participantes e a quantidade de números apostados por cada pessoa respectivamente. As pessoas apostam em inteiros maiores do que 1 e menores do que 50, inclusive. Cada uma das N linhas seguintes contém K números, representando as apostas de cada pessoa, uma pessoa por linha.

Saída

Imprima ‘S’ caso seja possível ou ‘N’ caso contrário.

Exemplos

Entrada 2 3 1 2 3 5 6 7	Saída S
Entrada 3 3 3 2 1 6 5 4 4 4 4	Saída S
Entrada 4 3 9 4 7 4 4 4 2 7 2 2 2 1	Saída N