
Seletiva para Maratona de Programação de 2018

Instituto de Matemática e Estatística
Universidade de São Paulo

Caderno de Problemas

Apoio e patrocínio:



www.caelum.com.br

Departamento de Ciência da Computação IME-USP

Sábado, 18 de agosto de 2018.

Instruções

- A competição tem duração de 5 horas;
 - Faltando 1 hora para o término da competição, o placar será congelado, ou seja, o placar não será mais atualizado;
 - Faltando 15 minutos para o término da competição, os times não receberão mais a resposta de suas submissões.
-
- 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;
 - Na saída, toda linha deve terminar com o caracter ‘\n’;
 - O nome do arquivo de códigos em Java deve ser **exatamente** como indicado abaixo do nome de cada problema. Para C/C++ é recomendado usar o nome indicado;
 - Para códigos em Java, o nome da classe principal deve ser **igual** ao nome do arquivo.

Respostas das submissões		
Not answered yet	-	Paciência
YES	-	Código aceito. Parabéns!
NO	Compilation error	Erro de compilação
	Wrong answer	Errado. Pode tentar de novo.
	Time limit exceeded	Seu programa demora muito para dar a resposta (certa ou errada)
	Runtime error	Erro em tempo de execução (ex.: <i>segmentation fault</i>)
	Problem name mismatch	Leia as duas últimas instruções
	Presentation error	Não está imprimindo no formato exigido no enunciado
	If possible, contact staff	Não sei, você conseguiu fazer algo inesperado

Problema A: Estudando curvas de nível

Arquivo: *curvas*. [c/cpp/java]

O Instituto de Geografia e Ordenamento do Território da Universidade de Lisboa (IGOT) está realizando várias pesquisas na Ilha do Pico. Uma dessas pesquisas se foca em estudar as curvas de nível da montanha do Pico, que é a mais alta montanha em Portugal. Usando programas de processamento de imagens, os pesquisadores têm transformado tais curvas em polígonos simples (que não se auto-intersectam).

Para entender como o terreno muda com relação à altura, o IGOT está interessado na *monotonicidade* de uma curva de nível com respeito a um conjunto de direções (vetores). Dado um vetor v , dizemos que um polígono simples é v -monótono se a intersecção de toda reta, cuja inclinação é ortogonal a v , é um segmento (que pode ser um ponto) ou é vazia. Você faz parte da equipe de desenvolvimento da IGOT, e eles lhe encarregaram a seguinte tarefa, dado um polígono simples e um conjunto de vetores, responder para cada vetor, se o polígono é monótono segundo esse vetor.

Entrada

A primeira linha contém dois inteiros N e Q , o número de pontos do polígono e o número de consultas, respectivamente. Cada uma das próximas $N+Q$ linhas contém dois inteiros. Dentre elas, a i -ésima linha representa o ponto (x_i, y_i) , se $i \leq N$, caso contrário, ela representa o vetor (v_x, v_y) . Os pontos do polígono são dados ordenados em sentido anti-horário.

Saída

Por cada consulta, imprima “Y” (sem aspas) se o polígono é monótono para o vetor correspondente, caso contrário imprima “N”.

Restrições

- $1 \leq N, Q \leq 10^5$
- $-10^9 \leq x_i, y_i, v_x, v_y \leq 10^9$
- É garantido que o polígono tem área positiva.

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
6 2	Y
4 0	N
4 2	
1 2	
1 4	
0 4	
0 0	
0 1	
3 2	

Exemplo de entrada	Saída para o exemplo de entrada
6 4	N
0 0	N
16 2	Y
0 10	N
-10 0	
0 -10	
20 -4	
10 10	
10 -10	
0 -20	
-20 10	

Na Figura 1 se mostra o polígono do primeiro exemplo e o vetor \overrightarrow{DG} que representa a segunda consulta.

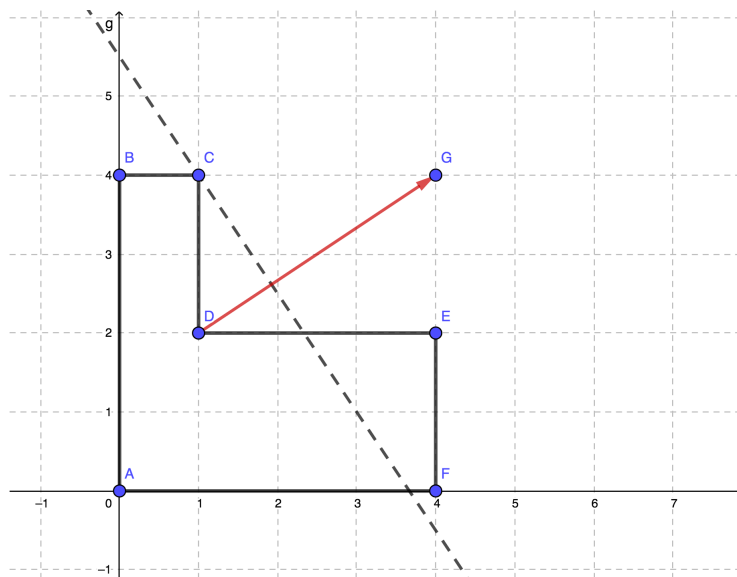


Figura 1: Polígono do primeiro exemplo

Problema B: Estética na poesia

Arquivo: *poesia*. [c/cpp/java]

Um dos grandes poetas da língua portuguesa foi o Luiz Vaz de Camões nascido em Lisboa em torno de 1524. As obras de Camões são leitura fundamental para os alunos de ensino médio em países como Brasil ou Portugal. As redondilhas (versos de cinco ou sete sílabas) e os sonetos (composições de catorze versos) de Camões são considerados por muitos a mais importante produção lírica em português de todos os tempos.

Camões foi um visionário para sua época, sendo considerado o grande modelo da língua portuguesa. Sua obra tem sido estudada desde distintos aspectos: histórico, cultural, simbólico, etc. O Carlitos “o eficiente” Nunes está estudando a estética na poesia de Camões. Ele está especialmente interessado em uma métrica baseada no tamanho dos versos de um poema. Definimos tal métrica a seguir. Suponha que temos um poema composto por N versos. Dizemos que um poema é K -elegante se $K \geq 2$, N é múltiplo de K e, além disso, existem exatamente N/K versos cujo tamanho ao ser dividido por K tem resto igual a i , para $i = 0, 1, \dots, K - 1$. Observe que um mesmo poema pode ser K -elegante para distintos valores de K .

Carlitos já processou os dados de distintos poemas de Camões e agora precisa sua ajuda para determinar a menor elegância de cada um dos poemas.

Entrada

A entrada é composta por duas linhas. A primeira linha contém um inteiro N , o número de versos de um poema. Na segunda linha são dados N inteiros, digamos $\ell_1, \ell_2, \dots, \ell_N$, tal que ℓ_i representa o tamanho do i -ésimo verso do poema.

Saída

Um único inteiro K , o menor inteiro tal que o poema é K -elegante. Se não existir tal inteiro, imprima “-1”.

Restrições

- $1 \leq N \leq 2 \cdot 10^3$
- $1 \leq \ell_i \leq 10^9$

Exemplos

Exemplo de entrada 6 3 6 1 7 8 14	Saída para o exemplo de entrada 2
Exemplo de entrada 5 17 21 14 35 13	Saída para o exemplo de entrada 5

Explicação

No primeiro caso, se $K = 2$, então 6, 8 e 14 (respectivamente, 3, 1 e 7) têm resto 0 (respectivamente, resto 1) quando divididos por K .

Problema C: Passeando pela lagoa

Arquivo: *lagoa*. [c/cpp/java]

A cidade do Porto deve receber as finais mundiais do ICPC no ano de 2019. Um dos pontos turísticos secretos da cidade é a chamada “lagoa das mil pontes”. O Sr. Manoel Pontes (incrível, é esse mesmo o nome dele ...) construiu essa beleza na lagoa em anos e anos de trabalho duro. A lagoa tem várias pequenas ilhas naturais. O Sr. Manoel construiu com o passar do tempo, pequenas pontes de madeira ligando as ilhas. Em alguns casos, há até mesmo mais de uma ponte ligando alguns pares de ilhas. Os visitantes da atração se divertem passeando pelas pontes, e conhecendo as pequenas ilhas. Além disso, passeando pelas pontes os visitantes conseguem ir de uma ilha para qualquer outra.

O Sr. Manuel está interessado em incrementar ainda mais a atração turística, a fim de atrair mais visitantes. Uma ideia que ele teve é propor jogos aos visitantes. Ele deseja propor o seguinte desafio: é possível sair do início do passeio (fora da lagoa), passar por todas as pontes, sem repetir nenhuma, e retornar ao ponto de início (ou seja, sair da lagoa)?

Antes de criar a atração, ele mesmo tentou, tentou, mas não conseguiu descobrir se era possível. Para tornar o desafio ainda mais interessante, o Sr. Manuel vai permitir adicionar certo número de pontes entre algumas ilhas da lagoa. Ele deseja saber, então, se adicionando algumas dessas pontes, aquele passeio que ele deseja propor se tornará possível. Sua tarefa neste problema é decidir se é possível escolher algumas (ou nenhuma) destas pontes que, se adicionadas, permitem um passeio como descrito.

Entrada

A primeira linha contém três inteiros N , M e K , onde N é o número de ilhas, M é o número de pontes iniciais e K é o número das pontes que é possível adicionar. As ilhas da lagoa são identificadas por inteiros distintos entre 1 e N . Cada uma das próximas $M + K$ linhas descrevem uma ponte. Cada uma dessas linhas contém dois inteiros a e b , que representam a existência ou a possível adição de uma ponte entre as ilhas a e b . As primeiras M linhas descrevem as pontes que já existem, e as K linhas seguintes descrevem as pontes que você pode adicionar.

Saída

Na primeira linha Imprima “YES” (sem aspas), se tal passeio é possível ou “NO” caso contrário. Caso exista solução, imprima na segunda linha um inteiro R , $0 \leq R \leq K$, que representa o número de pontes que precisamos adicionar. Depois imprima R linhas, cada uma contendo dois inteiros, as ilhas que são conectadas por essa ponte, em qualquer ordem. Se existem múltiplas soluções, qualquer uma será aceita.

Restrições

- $1 \leq N \leq 3 \cdot 10^5$
- $0 \leq M, K \leq 3 \cdot 10^5$
- $1 \leq a < b \leq N$
- É garantido que podemos ir de uma ilha até qualquer outra usando apenas as pontes iniciais.

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
4 3 4	YES
1 2	1
2 3	1 4
3 4	
1 4	
1 4	
1 3	
3 4	

Exemplo de entrada	Saída para o exemplo de entrada
4 3 2	NO
1 2	
2 3	
3 4	
1 2	
3 4	

Problema D: Maximizando a publicidade

Arquivo: *publicidade*. [c/cpp/java]

Em Portugal dois dos partidos políticos mais populares são o Partido Social Democrata (PSD) e o Partido Socialista (PS). As eleições presidenciais de Portugal são em 2019, e os partidos políticos estão trabalhando na melhor forma de fazer publicidade para atingir o maior número de pessoas.

Devido ao aumento do custo da publicidade eleitoral, o PSD e o PS desejam fazer um acordo e repartir os custos da publicidade. Os encarregados de fazer a negociação têm os dados da preferência política de algumas pessoas em Portugal. Eles têm filtrado a informação, obtendo assim, a posição geográfica das pessoas que simpatizam pelo PSD e o PS. Essa posição é representada por um ponto no plano cartesiano. Para fazer os cálculos dos gastos, eles desejam encontrar duas áreas disjuntas que, ao somar os simpatizantes do próprio partido, o resultado seja o maior possível. Para facilitar os cálculos, eles decidiram considerar ambas áreas como sendo retângulos com lados paralelos aos eixos. Em outras palavras, eles desejam encontrar dois retângulos, um representando o PSD e outro representando o PS, tal que a soma do número de simpatizantes do PSD no primeiro retângulo e do número de simpatizantes do PS no segundo retângulo seja máxima. Você foi contratado para desenvolver um programa que realizará tal tarefa.

Entrada

A primeira linha contém um inteiro N , o número de pessoas cujos dados serão considerados. As próximas N linhas descrevem os dados de cada pessoas. A i -ésima dessas N linhas contém dois inteiros e um caractere, digamos x_i , y_i e p_i . Os inteiros representam o ponto (x_i, y_i) e p_i representa a preferência política da i -ésima pessoa.

Saída

O valor máximo que se obtém ao somar o número de simpatizantes em cada retângulo. Neste problema consideramos que os pontos no perímetro fazem parte do retângulo.

Restrições

- $1 \leq N \leq 10^6$
- $-10^9 \leq x_i, y_i \leq 10^9$
- $p_i = \text{"b"}$ ou $p_i = \text{"w"}$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
3 0 0 b 0 1 b 1 1 b	3

Exemplo de entrada	Saída para o exemplo de entrada
<p>4 0 0 b 1 1 w 2 2 b 3 3 w</p>	<p>3</p>

Exemplo de entrada	Saída para o exemplo de entrada
<p>4 -1 -1 b 0 0 b 1 1 w 2 2 w</p>	<p>4</p>

Problema E: Trabalho em grupo

Arquivo: *grupo*. [c/cpp/java]

Estudos modernos de educação apontam nos benefícios pedagógicos de fazer trabalhos em grupo, especialmente nos últimos anos do ensino fundamental e início do ensino médio. Grupos diferentes, de vários tamanhos, são ótimos e permitem que os estudantes entendam as diferenças e treinem trabalhar com várias pessoas. Um professor de uma escola moderna está interessado em saber quantos grupos diferentes ele consegue formar em sua turma com N alunos. A única restrição é que um grupo deve ter pelo menos 2 alunos, e dois grupos são distintos se pelo menos um estudante está em um deles e não está no outro.

Sua tarefa neste problema é, dado N calcular o número de grupos diferentes que é possível formar na turma.

Entrada

Uma única linha contendo o inteiro N .

Saída

Uma única linha contendo o número de grupos diferentes que é possível formar na turma.

Restrições

- $1 \leq N \leq 30$

Exemplos

Exemplo de entrada 2	Saída para o exemplo de entrada 1
Exemplo de entrada 3	Saída para o exemplo de entrada 4

Explicação

No primeiro exemplo temos dois estudantes, portanto, somente podemos formar um grupo. No segundo exemplo temos três estudantes, digamos A , B e C . Com eles podemos formar os seguintes grupos: AB , AC , BC e ABC .

Problema F: Otimizando o transporte de Portugal

Arquivo: *transporte*. [c/cpp/java]

O Ministério das Obras Públicas, Transportes e Comunicações de Portugal está realizando planos para melhorar as vias de transporte em todo o país. A principal preocupação do ministério é o atraso que os cidadãos sofrem diariamente devido às vias interditadas. Parte fundamental do plano do ministério é identificar as vias cuja interdição causam maior prejuízo à população. Por tal motivo, eles contactaram o Instituto de Mobilidade Eficiente (IME).

O IME está desenvolvendo um software que dada a descrição das vias de transporte em Portugal e a informação sobre o uso de cada uma delas, encontra as vias que mais prejudicam à população em caso de uma interdição. O encarregado deste importante projeto é o Marcel “o otimizador” Saito. Atualmente Marcel está ocupado realizando tarefas administrativas e organizando a equipe de trabalho. Ele confia no seu potencial, por isso, encarregou você uma parte crucial desse software. O problema que você deve resolver é o seguinte: dados dois locais em uma cidade, digamos s e t , queremos saber quanto é o maior prejuízo de tempo que uma pessoa sofre, ao se deslocar de s para t , quando acontece a interdição de exatamente uma via da cidade. Para fins práticos, pode supor que as pessoas sempre escolhem um caminho que leva o menor tempo para se deslocar de um lugar para outro.

Entrada

A primeira linha da entrada contém dois inteiros, N e M , o número de locais da cidade e o número de vias que ligam esses locais, respectivamente. Os locais da cidade são representados pelos inteiros 1 até N . A segunda linha contém dois inteiros s e t . Cada uma das próximas M linhas contém três inteiros, digamos u , v e w , que descrevem uma via que liga o local u com o local v e cujo tempo de percurso é w . As vias são de duas mãos, ou seja, podem ser percorridas em qualquer direção. É garantido que sempre existe uma forma de chegar de s até t .

Saída

Um único inteiro, o maior tempo de um caminho mínimo de s para t quando interditamos exatamente uma via da cidade. Se existe alguma via cuja interdição desconecta s e t , imprima “-1” (sem aspas).

Restrições

- $1 \leq N \leq 10^5$
- $1 \leq M \leq 5 \cdot 10^5$
- $1 \leq u, v \leq N, u \neq v$
- $1 \leq w \leq 10^4$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
10 12 1 5 1 4 1 1 2 6 1 3 2 2 3 2 3 5 3 3 8 3 4 6 5 5 7 1 5 9 5 6 10 20 7 8 7 9 10 3	13

Problema G: Administrando um presídio

Arquivo: *presidio*. [c/cpp/java]

A penitenciária de Viana do Castelo é um dos maiores estabelecimentos prisionais existentes na Europa. O prédio dispõe de milhares de celas, dispostas linearmente em vários andares, que são numerados de forma que celas de números adjacentes são também vizinhas, seja uma ao lado da outra, ou uma em cima da outra, ligadas por uma escada adjacente. Não se sabe bem porque, mas as celas são numeradas a partir de um número negativo, de forma que a cela de número zero fica bem no meio do presídio, exatamente no centro do andar central do prédio. Provavelmente o primeiro diretor, que decidiu a numeração das celas, fez isso para se divertir ...

Cada guarda do presídio recebe um intervalo de celas para cuidar. Ele deve fazer sua ronda entre estas celas, verificar que os presos não fugiram, estão bem, etc. O diretor do presídio, a partir dos intervalos de celas atribuídas a cada um dos guardas faz dois tipos de operações:

1. Dado um guarda i , mudar o intervalo das celas atribuídas a este para $[\ell, r]$;
2. Dado um intervalo de guardas $[a, b]$, descobrir quantas celas são verificadas por todos os guardas deste intervalo naquele instante.

Sua tarefa neste problema é ler o número N de guardas do presídio e, para cada guarda, o intervalo de celas que estará sob sua responsabilidade. Em seguida, você deverá ler as Q operações realizadas pelo diretor do presídio, efetuando se for uma alteração do tipo 1, ou respondendo a consulta, se for do tipo 2.

Entrada

A primeira linha contém dois inteiros N e Q , o número de guardas e o número de operações a ser realizadas, respectivamente. As próximas N linhas descrevem a informação de um guarda. A i -ésima dessas linhas contém dois inteiros L_i e R_i , o intervalo de celas atribuídas ao i -ésimo guarda. Finalmente, as Q linhas seguintes descrevem as operações feitas pelo diretor. Cada uma dessas linhas representa uma operação da seguinte maneira.

- “C i ℓ r ”: que indica mudar o intervalo de celas do i -ésimo guarda para $L_i = \ell$ e $R_i = r$.
- “? a b ”: que indica, dado o intervalo de guardas $[a, b]$, descobrir quantas celas são verificadas por todos os guardas nesse intervalo.

Observe que a cada instante, um guarda tem um único intervalo de celas atribuído a ele.

Saída

Para cada operação do tipo ‘?’, imprima uma linha contendo a resposta.

Restrições

- $1 \leq N, Q \leq 2 \cdot 10^5$
- $-10^9 \leq L_i \leq R_i \leq 10^9$
- Para cada operação do tipo ‘C’, $1 \leq i \leq N$ e $-10^9 \leq \ell \leq r \leq 10^9$

- Para cada operação do tipo '?', $1 \leq a \leq b \leq N$
- Existe pelo menos uma operação do tipo '?'

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
3 4	4
1 100	4
5 10	1
7 20	
? 1 3	
? 2 3	
C 1 8 8	
? 1 3	

Exemplo de entrada	Saída para o exemplo de entrada
2 3	2
1 2	2
9 10	0
? 1 1	
? 2 2	
? 1 2	

Problema H: Produção de vinhos

Arquivo: *vinho*. [c/cpp/java]

A cidade do Porto é conhecida mundialmente pela produção de vinhos licorosos. Estes vinhos são produzidos nas regiões próximas da cidade, e em seguida envelhecidos em barris de carvalho atingindo o sabor inconfundível que é apreciado em todo o mundo. As vinícolas precisam seguir padrões rigorosos na produção para receberem o selo de qualidade que garante sua venda no mercado. Em particular, a temperatura nos parreirais é medida diariamente, e deve, segundo os estudos dos especialistas, para atingir os mais altos padrões de qualidade, respeitar especificações muito precisas.

Para que a uva atinja seu melhor desenvolvimento, é importante saber quantas temperaturas diferentes tivemos durante o seu crescimento, e em quantos dias cada uma destas temperaturas se repetiu. Estudos recentes mostram que a qualidade da uva estará intimamente ligada ao seguinte número. Considere o período de crescimento da uva, e conte, nestes dias, quantas temperaturas diferentes tivemos e, para cada temperatura, quantas vezes ela se repete no período. Dizemos que a produção de uma parreira terá qualidade x se tivermos pelo menos x temperaturas diferentes que se repetiram em x dias durante o crescimento das suas uvas.

Sua tarefa será calcular a qualidade da produção de cada parreira da vinícola. São dadas as temperaturas medidas em N dias no campo. Em seguida, para cada uma das Q parreiras da vinícola são dados o dia do início e do fim do crescimento de suas uvas. Para cada uma destas parreiras você deve calcular a máxima qualidade de sua produção.

Entrada

A primeira linha contém dois inteiros N e Q , o número de dias em que a temperatura foi medida no campo e o número de parreiras, respectivamente. A segunda linha contém N inteiros separados por um espaço. O i -ésimo desses inteiros indica a temperatura medida no dia i . Finalmente, as seguintes Q linhas descrevem cada uma das parreiras. A i -ésima dessas linhas contém dois inteiros, ℓ_i e r_i , que indicam o dia do início e do fim do crescimento das uvas da i -ésima parreira.

Saída

Imprima Q linhas, a i -ésima dessas linhas deve conter a máxima qualidade da parreira i .

Restrições

- $1 \leq N, Q \leq 2 \cdot 10^4$
- A temperatura de cada dia está entre -10^9 e 10^9
- $1 \leq \ell_i \leq r_i \leq N$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
6 3	2
1 2 3 1 2 1	1
1 6	2
2 4	
1 5	

Explicação

Na primeira parreira foram medidas 3 temperaturas distintas. Somente uma se repete pelo menos três vezes e duas se repetem pelo menos duas vezes, portanto, a máxima qualidade da parreira é 2.

Na segunda parreira as temperaturas medidas foram todas distintas, logo a máxima qualidade dela é 1. Finalmente, a máxima qualidade da terceira parreira é 2, já que temos duas temperaturas que se repetem duas vezes.

Problema I: Uma história sobre o chá

Arquivo: *cha*. [c/cpp/java]

O chá é a segunda bebida mais consumida no mundo após a água, especialmente no Reino Unido, onde em torno de 84% da população o consome diariamente. É um fato conhecido a tradição que os britânicos têm por esta bebida. O que poucos conhecem é que o chá inglês teve origem em Portugal. Em 1662, quando Catarina de Bragança casou o Rei Carlos II de Inglaterra, viajou para Londres levando entre seus pertences folhas de chá. Diz a lenda que as caixas, onde essas folhas eram transportadas, foram marcadas como “Transporte de Ervas Aromáticas”, mais tarde abreviado para TEA.

A historiadora Gaia “a curiosa” Fontenelle está investigando alguns documentos antigos que relatam como o chá era transportado de Portugal até Inglaterra. Inicialmente, N barcos estavam ancorados em Portugal. Cada um deles era atribuído com um inteiro distinto entre 1 e N . Os documentos afirmam que, para transportar o chá, se faziam em total K viagens. Nessas viagens estavam envolvidos os portos de Portugal, China e Inglaterra. Após fazer muitas indagações, Gaia chegou nas seguintes conclusões:

1. Em cada uma das K viagens somente um barco se trasladava de um porto a outro porto. Além disso, aquele barco tinha o menor número se consideramos os barcos ancorados em **ambos** os portos.
2. As viagens aconteciam de forma sequencial, nunca existiam dois barcos viajando ao mesmo tempo.
3. O barco N terminava seu itinerário quando chegava em Inglaterra. Para $i < N$, o barco i só terminava seu itinerário quando chegava em Inglaterra e o barco $i + 1$ tinha terminado seu itinerário também.

Gaia percebeu rapidamente que, com essas regras, cada porto se assemelha a uma pilha, onde uma viagem é equivalente a mover o menor barco (topo) da pilha origem para a pilha destino (respeitando a restrição 1.). Acostumada a lidar com fontes de informação imprecisa ou falsa, começou a brincar no seu caderno, e percebeu que dependendo dos valores de N e K , as vezes trasladar os N barcos era impossível (mesmo usando o porto da China). Por tal motivo, para cada relato sobre o transporte do chá, ela quer determinar se ele é possível ou não.

Gaia sentiu fome e saiu para um tomar café, comer uns chocolates e fotografar o pôr do sol. Você é um estagiário da equipe dela e deseja impressioná-la (talvez desta forma consiga o emprego). Assim, decidiu resolver o problema antes de ela voltar.

Entrada

A entrada é composta por uma única linha contendo os inteiros N e K , o número de barcos e o número de viagens totais feitas pelos barcos, respectivamente.

Saída

Na primeira linha imprima “Y” (sem aspas) se é possível trasladar os N barcos de Portugal até Inglaterra, respeitando as restrições dadas no enunciado. Caso contrário, imprima “N”. Caso a resposta seja afirmativa, imprima K linhas que descrevam um itinerário de viagens que traslada

os N barcos. Para representar uma viagem do porto X ao porto Y , imprima “ $X Y$ ”. Cada porto é representado por um caractere da seguinte forma:

- Portugal: “A”
- China: “B”
- Inglaterra: “C”

Se existe mais de um possível itinerário de viagens, qualquer um será aceito.

Restrições

- $1 \leq N \leq 21$
- $1 \leq K \leq 2^{21}$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
2 3	Y A B A C B C

Exemplo de entrada	Saída para o exemplo de entrada
2 1	N

Exemplo de entrada	Saída para o exemplo de entrada
2 6	Y A B B A A C C B A C B C

Explicação

A saída do primeiro exemplo descreve o seguinte itinerário:

1. O barco 1 viaja de Portugal a China.
2. O barco 2 viaja de Portugal a Inglaterra.
3. O barco 1 viaja de China a Portugal.

Problema J: Guerra dos Memes

Arquivo: *memes*. [c/cpp/java]

Os desencontros entre Brasil e Portugal tem uma longa história. A mais recente discrepância aconteceu em 2016 quando estes países tiveram uma “guerra” de memes. A jornalista Fiorentina “a sagaz” Azzellini está estudando este acontecimento como parte de um trabalho que tenta explicar como as redes sociais têm mudado a forma de interação entre as pessoas.

O Brasil avassalou a internet com uma grande quantidade de memes, ganhando esta disputa em três dias. A Fiorentina, investigando como puderam ser criados tal quantidade de imagens, descobriu que uma parte delas tinha uma construção muito simples. O meme era composto por uma imagem que envolvia as bandeiras de Brasil e Portugal, e uma palavra. A palavra $S(x)$ era criada a partir de uma letra x , e tem a seguinte regra de formação

$$S(x) = \begin{cases} 'a', & \text{se } x = 'a', \\ S(p(x)) \cdot x \cdot S(p(x)), & \text{caso contrário,} \end{cases}$$

onde $p(x)$ é a letra que precede x no alfabeto e $A \cdot B$ denota a concatenação das palavras A e B . A Fiorentina começou a brincar com distintas letras e escreveu no seu caderno

$$\begin{aligned} S('b') &= 'aba', \\ S('c') &= 'abacaba', \\ &\vdots \end{aligned}$$

Com a sagacidade que a caracteriza, a Fiorentina rapidamente percebeu que a palavra $S('z')$ é enorme. Sem muita vontade de escrever ela no caderno, mas querendo saber algumas letras dessa palavra, ela pediu a sua ajuda. A Fiorentina gostaria de saber qual é a letra na n -ésima posição da palavra $S('z')$.

Entrada

Um único inteiro n .

Saída

Um único caracter, a letra na n -ésima posição da palavra $S('z')$.

Restrições

- $1 \leq n \leq 10^3$

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
1	a

Problema K: Brincadeiras portuguesas

Arquivo: *brincadeira*. [c/cpp/java]

As brincadeiras de infância portuguesas são muito sofisticadas. Algumas destas brincadeiras têm regras complicadas, e sua descrição pode deixar confuso até mesmo um adulto, mas, de alguma forma, as crianças lá brincam e se divertem muito com elas. Uma delas é a brincadeira do “roda-e-gira”. Nesta, um dos participantes anota num papel uma ordem dos participantes, que fica em segredo dos outros. Um dos outros participantes tenta então descobrir esta ordem fazendo o que eles chamam de roda-e-gira. A ideia é aplicar várias vezes uma mesma permutação dos participantes, a partir da ordem de idade das crianças. Se, em algum destes passos, a permutação secreta for descoberta, a criança ganhou. Caso contrário, a criança que imaginou a ordem secreta é a vencedora da brincadeira. As crianças de Portugal dizem que este é um jogo “gira”¹.

Vamos dar um exemplo. Imagine, por exemplo, que a ordem secreta escolhida seja $(4, 2, 1, 3)$, ou seja, a criança mais velha em primeiro, depois a segunda mais jovem, a mais jovem e por fim a terceira. O primeiro participante poderia tentar a permutação $(3, 1, 4, 2)$. Aplicando a permutação a partir da ordem de idade dos participantes $(1, 2, 3, 4)$ obtemos $(3, 1, 4, 2)$. Aplicando novamente sobre esta ordem, $(4, 3, 2, 1)$. Novamente, $(2, 4, 1, 3)$ e mais uma vez voltamos à ordem inicial $(1, 2, 3, 4)$. Com isso, este participante perdeu o jogo. O participante que imaginou a ordem secreta ganhou. No jogo as crianças ficam se movendo, rodando e girando, daí o nome . . .

Como vemos, é um jogo bastante sofisticado, e intrigou um famoso matemático português que resolveu estudá-lo. Sua tarefa neste problema é ajudar este matemático. Dada uma ordem secreta p dos participantes da brincadeira, desejamos descobrir quantas permutações q podem resultar em p quando aplicadas um número finito $k \geq 0$ de vezes, como no exemplo acima.

Entrada

A primeira linha contém dois inteiros N e k , o número de crianças e as vezes que desejamos aplicar uma permutação q para obter p , respectivamente. A segunda linha contém N inteiros separados por um espaço em branco, a ordem representada pela permutação p .

Saída

Imprima um inteiro, o número de permutações q que, quando aplicadas k vezes, resultam na permutação p . Imprima esse número módulo $10^9 + 7$.

Restrições

- $1 \leq N \leq 10^5$
- $0 \leq k \leq 10^6$

¹Gíria em Portugal para interessante, bonito, divertido

Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
5 60 1 2 3 4 5	120

Exemplo de entrada	Saída para o exemplo de entrada
5 2 3 5 1 4 2	2