



Seletiva para Maratona de Programação de 2024

Instituto de Matemática e Estatística
Universidade de São Paulo

Caderno de Problemas

Apoio e patrocínio:

alura

Departamento de Ciência da Computação IME-USP
Sábado, 10 de Agosto de 2024.

Instruções

Regras

- A competição tem duração de 5 horas;
- Os times são compostos de no máximo 3 integrantes;
- Os times podem utilizar somente um computador;
- Faltando 1 hora para o término da competição, o placar será congelado, ou seja, o placar não será mais atualizado;

Entrada e Saída

- 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 caractere ‘\n’.

Índice de Problemas

- A. Nauryz
- B. Derrubando árvores
- C. Ciclismo de estrada
- D. M de Maçã
- E. Crise energética
- F. Carbono neutro
- G. Teleportando pelo Cazaquistão
- H. Semáforo
- I. De Baikonur a Marte
- J. Acarajé
- K. Pegando pelúcias
- L. Noite em Hazrat Sultan

Problem A. Nauryz

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

Alikhan está preparando uma festa de ano novo persa. A atração principal da festa será um aparelho onde os convidados podem colocar uma música na fila de reprodução que será tocada para a festa inteira.

Durante a festa, o convidado i vai até o aparelho no tempo t_i e escolhe uma música de duração m_i que será adicionada no final da fila de reprodução.

O aparelho só tem um problema, o convidado pode apertar um botão ao escolher a música e furar a fila, interrompendo a música que está sendo tocada naquele momento e iniciando imediatamente a reprodução da música escolhida. A música interrompida não é tocada novamente. Felizmente, caso isso aconteça, o restante da fila permanece igual.

Alikhan sabe que se a música do convidado i estiver tocando e for substituída, este convidado vai ficar muito triste e nunca mais vai aparecer em nenhuma festa de Alikhan (note que, se alguém apertar o botão no instante em que a música do convidado i terminar, este convidado não irá ficar triste). Para remediar essa situação, Alikhan precisa saber todos os convidados que tiveram suas músicas interrompidas para que ele possa mandar uma carta de desculpas no dia seguinte.

Input

A primeira linha contém um único inteiro $1 \leq n \leq 10^5$, a quantidade de convidados que colocaram músicas no aparelho. Cada convidado interage com o aparelho apenas uma vez.

Cada uma das próximas n linhas contém três inteiros t_i, m_i, c_i ($1 \leq t_i, m_i \leq 10^7, c_i \in \{0, 1\}$), o tempo em que o convidado i interage com o aparelho, a duração da música, e se ele usou ou não o botão para furar a fila, respectivamente. É garantido que dois convidados não interagem com o aparelho ao mesmo tempo. O valor $c_i = 0$ caso o botão não foi usado pelo convidado i , $c_i = 1$ caso contrário.

Output

A saída deve conter duas linhas. A primeira linha deve conter f , a quantidade de convidados que ficaram tristes. A segunda linha deve conter f números, o índice dos f convidados que ficaram tristes, separados por espaço. Qualquer ordem desses índices será aceita.

Examples

standard input	standard output
3 1 4 0 2 2 0 6 5 1	1 2
2 1 5 1 6 5 1	0
4 1 2 0 20 1 1 3 5 0 2 20 1	2 1 4

Problem B. Derrubando Árvores

Input file: `standard input`
Output file: `standard output`
Time limit: 5 seconds
Memory limit: 256 megabytes

Mori cansou de morar no Brasil e se mudou diretamente para o Cazaquistão buscando uma vida melhor. No Cazaquistão ser programador é coisa do passado e a nova profissão do hype no momento é ser lenhador. Como Mori não é nenhum bobo, resolveu viver a vida de lenhador no Cazaquistão também!

O novo trabalho de lenhador de Mori estipula cortar M árvores distribuídas entre os pontos inteiros 1 a N . Como tudo é harmonioso no mundo 2D, todas as árvores tem altura H e tem suas raízes no eixo- x , isto é, $y = 0$. Não existem duas árvores no mesmo ponto.

Quando Mori derruba uma árvore localizada no ponto a_i , ele pode escolher por derrubá-la para esquerda, atingindo os pontos x tal que $a_i - H \leq x \leq a_i$, ou derrubá-la para direita, atingindo os pontos x tal que $a_i \leq x \leq a_i + H$. Se houver outras árvores no intervalo atingido, isto acarretará em uma série de eventos em cadeia, derrubando estas árvores na mesma direção em que foi derrubada a árvore original. Após terem sido derrubadas todas as árvores decorrentes de derrubar a árvore localizada no ponto a_i , Mori remove estas árvores e segue o processo de derrubar árvores com as árvores restantes.

Mori tem um empecilho ao realizar seu trabalho. O seu contratante não possui o terreno referente aos pontos 0 e $N + 1$. Então, durante o seu trabalho, ele não pode derrubar nenhuma árvore nesses dois pontos.

Como Mori não sabe ainda em quais pontos estão localizadas as M árvores, Mori quer saber a probabilidade de conseguir executar seu trabalho sem derrubar árvores nos pontos 0 e $N + 1$, dado que toda a configuração de M árvores espalhadas em N pontos são equiprováveis. Imprima a resposta módulo 998244353.

Input

Cada teste consiste em múltiplos casos de teste. A primeira linha da entrada contém um único número inteiro t ($1 \leq t \leq 10^4$) — o número de casos de teste. Então segue-se as descrições dos casos de teste.

Cada caso de teste contém uma única linha de entrada contendo três números inteiros N, M, H ($1 \leq N, H \leq 10^5, 1 \leq M \leq N$), representando respectivamente o número de pontos em que as árvores podem estar localizadas, o número de árvores e a altura das árvores.

É garantido que a soma de N sobre todos os casos de teste é menor ou igual a 10^5 .

Output

Para cada caso de teste, imprima um único número inteiro — a probabilidade de Mori conseguir completar seu trabalho módulo 998244353.

Formalmente, a probabilidade pode ser expressa como uma fração irredutível $\frac{x}{y}$. Você deve imprimir o valor $x \times y^{-1} \bmod 998244353$, onde y^{-1} é um número inteiro tal que $y \times y^{-1} \bmod 998244353 = 1$.

Example

standard input	standard output
5	1
2 1 1	0
2 1 2	499122177
4 2 2	0
2 2 2	250691060
55555 44444 333	

Problem C. Ciclismo de estrada

Input file: **standard input**
Output file: **standard output**
Time limit: 1.5 seconds
Memory limit: 256 megabytes

O ciclismo de estrada é um dos esportes em que os atletas cazaques mais se destacam. Assim, para estreitar laços com o Cazaquistão, a Sociedade Brasileira de Ciclismo (SBC) decidiu organizar uma maratona de ciclismo em território cazaque!

Com a intenção de interessar a maior quantidade de competidores de todos os níveis, a maratona de organizada pela SBC é diferente das usuais. O circuito tem n postos em que os ciclistas podem parar para descansar, beber um energético de graça e continuar. Os postos estão distribuídos de forma circular. Em outras palavras, para $1 \leq i < n$, o posto $i + 1$ encontra-se imediatamente depois do posto i e o posto 1 depois do posto n . A bebida energética que pode ser usada no posto i confere ao ciclista b_i unidades de energia. A quantidade de energia necessária para ir do posto i ao próximo é de c_i unidades. Desta forma, um ciclista com x unidades de energia pode avançar do posto i ao seguinte somente se $x \geq c_i$. Ademais, ele perde c_i unidades de energia ao fazer isso.

A SBC está testando diferentes cenários para decidir como organizar melhor a maratona. A sua tarefa é ajudar eles respondendo 3 tipos diferentes de consultas:

- 1 i : Suponha que um ciclista com energia inicial 0 começa o seu percurso no posto i . Diga qual é o último posto a que consegue chegar antes de parar. Se o ciclista puder continuar indefinidamente, imprima -1 .
- 2 i x : Atribua o valor x para b_i .
- 3 i x : Atribua o valor x para c_i .

Input

A primeira linha da entrada contém dois números inteiros n e q ($1 \leq n, q \leq 10^5$), o número de postos e o número de consultas, respectivamente.

A segunda linha contém n inteiros, os valores de b_i ($1 \leq b_i \leq 10^9$).

A terceira linha contém n inteiros separados, os valores de c_i ($1 \leq c_i \leq 10^9$).

As seguintes q linhas descrevem as consultas no formato explicado no enunciado. É garantido que o índice i está entre 1 e n e que o valor atribuído x é tal que $1 \leq x \leq 10^9$.

Output

Para cada consulta do tipo 1, imprima o último posto a que o ciclista consegue chegar antes de parar ou -1 se ele puder continuar indefinidamente.

Example

standard input	standard output
5 4	2
3 2 5 8 1	5
2 3 4 7 6	-1
1 2	
1 1	
3 4 2	
1 1	

Problem D. M de Maçã

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

As maçãs se originaram na região das Montanhas Tian Shan, no Cazaquistão. O nome da maior cidade do país, Almaty, é frequentemente traduzido como “cheia de maçãs”. Depois de descobrir esse fato interessante, Harada e Teos, os maiores fãs de maçãs do mundo, decidiram viajar para Almaty. Agora eles precisam da sua ajuda para preparar a bagagem!

Eles levarão apenas uma mala, que pode ser vista como uma caixa 3D que vai de $(0, 0, 0)$ a (x, y, z) . Cada um deles colocará apenas um objeto na mala, uma maçã! Teos já colocou sua maçã dentro da mala. A maçã de Teos pode ser vista como uma esfera 3D com centro em (tx, ty, tz) e raio r . Embora a maçã possa estar “flutuando” no ar, é garantido que ela está completamente contida dentro da mala. Agora Harada se pergunta qual é a maior maçã, ou seja, uma esfera 3D, que pode caber dentro da caixa. Claramente, as duas maçãs podem se tocar, mas não devem se intersectar em nenhum ponto.

Input

A primeira linha contém três números inteiros $1 \leq x, y, z \leq 10^9$, as dimensões da mala deles. A segunda linha contém três números de ponto flutuante $0 \leq tx, ty, tz \leq 10^9$, as coordenadas do centro da maçã de Teos. A terceira linha contém um número de ponto flutuante $10^{-9} \leq r \leq 10^9$, o raio da maçã de Teos. É garantido que a maçã de Teos está completamente contida dentro da mala.

Output

Saída um único número de ponto flutuante: o raio da maior esfera 3D que pode ser colocada dentro da mala. A resposta será considerada correta se o erro absoluto ou relativo for menor ou igual a 10^{-6} .

Recomendamos que a resposta seja impressa com pelo menos 15 casas decimais. Para isso use:

- `printf("%.15lf\n", res);` em C;
- `cout << setprecision(15) << fixed << res << endl;` em C++;
- `print("%.15f"% res)` em Python;

Examples

standard input	standard output
2 1 1 0.500000000 0.500000000 0.500000000 0.500000000	0.5000000000000000
1000000000 1000000000 1000000000 500000000.0 500000000.0 500000000.0 500000000.0	133974596.215561360120773
10 10 10 3.14159265 2.71828182 5.00000000 2.50000000	3.226106436260601

Note

No primeiro caso de teste, Harada pode colocar uma maçã com raio 0.5 em $(1.5, 0.5, 0.5)$. Qualquer maçã com raio maior que 0.5 não pode ser colocada na mala sem violar as restrições do problema.

Problem E. Crise energética

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

A União Europeia vive uma crise energética. Tendo em vista esse panorama, o Ministério de Energia do Cazaquistão deseja reformular a rede de caminhos que conecta as usinas de energia do país. O Cazaquistão possui N usinas espalhadas por todo o seu território. O Ministério tem a informação de todas as rotas que ligam essas usinas, além do custo envolvido na manutenção de cada rota. Em particular, o custo de manter a i -ésima rota varia ao longo do tempo e depende de três parâmetros a_i , b_i e c_i , seguindo a função

$$a_i t^2 + b_i t + c_i,$$

onde t representa o tempo.

Para realizar essa tarefa, o ministro tem contratado o Marcel “o otimizador” Saito. Após analisar a geografia do Cazaquistão, Marcel já tem um plano para reformular a rede de rotas do país. Porém, como isso implicaria fazer grandes mudanças, ele precisa mostrar quão bom é o seu plano. Para isso, ele deseja calcular qual é (foi ou será) o menor custo de conectar todas as usinas de energia usando as rotas existentes em um mesmo tempo t .

Marcel está muito ocupado preparando o seu mais novo aprendiz, o Daniel “o dinâmico” Ito. Por isso, ele encarregou essa tarefa a você.

Input

A entrada contém $M + 1$ linhas. A primeira linha contém dois inteiros N ($2 \leq N \leq 100$) e M ($N - 1 \leq M \leq 200$), o número de usinas de energia e o número de rotas entre estas usinas, respectivamente.

As seguintes M linhas contêm, cada uma, cinco inteiros x_i , y_i , a_i ($1 \leq a_i \leq 10^4$), b_i ($-10^4 \leq b_i \leq 10^4$) e c_i ($-10^4 \leq c_i \leq 10^4$) que indicam a existência de uma rota que liga as usinas x_i e y_i e cujo custo ao longo do tempo é $a_i t^2 + b_i t + c_i$. É garantido que:

- $1 \leq x_i, y_i \leq N$, $x_i \neq y_i$,
- $a_i t^2 + b_i t + c_i \geq 0$, para todo $t \in \mathbb{R}$,
- usando as rotas existentes, pode-se ir de uma usina para qualquer outra.

Além disso, existe no máximo uma rota direta entre quaisquer par de usinas.

Output

Uma única linha contendo o custo mínimo de conectar essas usinas ao longo do tempo. Sua resposta é considerada correta se o erro absoluto ou relativo não excede 10^{-6} .

Example

standard input	standard output
3 3 1 2 3 2 1 2 3 4 -3 10 1 3 1 -5 7	7.437500000

Problem F. Carbono neutro

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

Cazaquistão é um país que leva muito a sério o problema do aquecimento global. Por tal motivo, o governo cazaque está implementando a Economia Verde no país. Para isso, é necessário que todas as empresas do país atinjam a neutralidade de carbono.

Há n empresas em Cazaquistão e podemos modelar as transações entre elas com um grafo orientado com m arcos. Um arco que sai de u para v indica uma transação feita da empresa u à empresa v . O governo cazaque precisa tomar uma decisão para cada uma dessas transações, a qual pode ser modelada como a atribuição de um peso inteiro $-1 \leq w \leq 1$ ao arco correspondente. O peso $w = 0$ indica que a transação é proibida. Os pesos $w = -1$ e $w = 1$ indicam que a transação absorve ou emite carbono, respectivamente.

Para uma empresa v alcançar a neutralidade de carbono, deve cumprir-se que $c^+(v) = c^-(v)$, onde $c^+(v)$ indica a soma dos pesos dos arcos que saem de v e $c^-(v)$, a soma dos pesos dos que entram. É claro que proibindo todas as transações, o governo cazaque conseguiria sua meta de Economia Verde; contudo, as transações importantes não podem ser proibidas porque a economia do país colapsaria! Uma transação i é importante se $t_i = 1$.

Dada descrição das transações entre as empresas do Cazaquistão, diga se é possível implementar a neutralidade de carbono em todas as empresas. Se for possível, mostre uma solução.

Input

A primeira linha da entrada contém dois números inteiros n ($1 \leq n \leq 10^6$) e m ($1 \leq m \leq 10^6$).

As seguintes m linhas descrevem os arcos do grafo. A i -ésima delas descreve o i -ésimo arco com 3 números inteiros: u_i, v_i, t_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $0 \leq t_i \leq 1$). O valor $t_i = 1$ indica que o i -ésimo arco representa uma transação importante, que não pode ser proibida.

Output

Imprima uma linha com a letra “S” se existe solução ou “N” se não existe. Caso exista solução, na segunda linha, imprima m inteiros. O i -ésimo desses inteiros deve ser o peso atribuído ao i -ésimo arco.

Examples

standard input	standard output
4 5 1 2 1 1 3 0 3 2 0 2 4 0 3 4 1	S -1 1 0 -1 1
2 1 1 2 1	N
3 2 1 3 1 1 3 0	S -1 1

Problem G. Teleportando pelo Cazaquistão

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

As finais mundiais do ICPC em Astana estão chegando e sua equipe preparou uma lista de lugares imperdíveis para visitar! Sua jornada começa em sua casa e se estende por várias cidades e pontos turísticos importantes nas vastas e belas paisagens do Cazaquistão antes de chegar ao local da competição em Astana. Ao longo do caminho, você passará por várias paradas essenciais (posições) que deve visitar, incluindo a movimentada cidade de Almaty, a histórica cidade de Shymkent e as maravilhas perto de Karaganda.

A ICPC, ciente de que os competidores têm muitos lugares para visitar, criou um dispositivo especial de teletransporte para ajudar as equipes a cumprirem seus compromissos turísticos a tempo para a competição. Você pode viajar para a próxima parada normalmente ou usar o dispositivo especial de teletransporte. Cada vez que você usa um teletransporte, ele é destruído, e um novo ponto de teletransporte é criado em sua última parada antes de teletransportar.

Para simplificar o planejamento, considere que sua casa e o primeiro teletransporte estão no ponto 0 e os pontos turísticos estão dispostos em uma linha reta. O custo da viagem normal é a diferença absoluta na distância entre sua parada atual e a próxima parada. O custo de um teletransporte é a diferença absoluta na distância entre sua última posição de teletransporte e a próxima parada.

Sua tarefa é determinar o custo total mínimo de viagem para chegar ao local da competição em Astana a partir de sua casa usando o dispositivo de teletransporte. Note que seus colegas de equipe colocaram muito cuidado e esforço na elaboração deste itinerário e querem visitar os pontos turísticos em ordem.

Input

A primeira linha contém um inteiro positivo n ($1 \leq n \leq 1000$), o número de pontos turísticos.

A próxima linha contém n inteiros positivos a_i ($1 \leq a_i \leq 10^9$), separados por espaço, que são as coordenadas dos pontos turísticos.

Output

Um único inteiro, a distância total mínima para visitar todos os pontos em ordem.

Examples

standard input	standard output
3 10 4 12	16
7 1 2 1 1 2 2 1	3

Note

A solução para o primeiro caso teste é a seguinte:

- a equipe e o teletransporte começam no ponto de coordenada 0;
- a equipe visita o ponto na coordenada 10 (+10);
- a equipe se teletransporta para o ponto na coordenada 0 (0+);
- (Um teletransporte é criado no ponto 10 e o da coordenada 0 é destruído);
- a equipe visita o ponto na coordenada 4 (+4);

- a equipe se teletransporta para o ponto na coordenada 10 (0+);
- (Um teletransporte é criado no ponto 4 e o da coordenada 10 é destruído);
- a equipe visita o ponto na coordenada 12 (+2).

Problem H. Semáforo

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

Nathan, como sempre, foi comprar energético para participar de uma competição em Astana, Cazaquistão. Como o povo cazaquistânês é saudável, energético se tornou uma raridade. Por tal motivo, Nathan tem que se deslocar para a única loja de energéticos na cidade de carro e então se locomover para o local de competição. Nathan, como sempre, não se planejou adequadamente, e ao chegar à loja de energéticos notou que pode não ter mais tempo de voltar para a competição. Ajude Nathan saber o menor tempo possível que ele consegue chegar na competição dado que ele está na loja de energéticos.

A cidade pode ser descrita por um conjunto de M ruas e N intersecções. Cada rua r_i , por sua vez, é descrita por duas intersecções, que representa as extremidades da rua, e um semáforo. Um semáforo permite ou impede de um carro passar pela rua que o contém caso ele esteja aberto ou fechado. Os semáforos, por sua vez, tem um comportamento cíclico e conhecido por todos na cidade. O semáforo da rua r_i cicla, ficando x_i minutos abertos e y_i minutos fechados. Todo semáforo no tempo 0 inicia seu ciclo.

A loja de energéticos se localiza na intersecção 1, enquanto a competição acontece na intersecção N , qual o menor tempo que Nathan consegue chegar na competição caso ele esteja na loja de energéticos no tempo t ?

A espera nos semáforos é tanta que o tempo percorrido andando com o carro é desprezível. Como Nathan não é tão incapaz ao ponto de usar um meio de transporte que é impossível chegar na competição, se tem a certeza que é possível chegar a intersecção N iniciando-se na intersecção 1.

Input

A primeira linha de entrada contém três números inteiros N , M e t ($2 \leq N \leq 10^5$, $N - 1 \leq M \leq 2 \times 10^5$, $1 \leq t \leq 10^9$), representando respectivamente o número de intersecções, o número de ruas e o tempo em que Nathan sairá da loja de energéticos.

Cada uma das próximas M linhas da entrada contém quatro números inteiros a_i , b_i , x_i , y_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$, $1 \leq x_i, y_i \leq 10^9$), representando que a rua r_i conecta as intersecções a_i e b_i e contém um semáforo que cicla ficando x_i minutos aberto e y_i minutos fechado.

É garantido que existe um caminho entre a intersecção 1 e a intersecção N . Não existe duas ruas que conectem as mesmas duas intersecções.

Output

Imprima um único número inteiro — o menor tempo em que Nathan consegue chegar à cidade N .

Examples

standard input	standard output
2 1 3 1 2 4 4	3
2 1 3 1 2 3 4	7
4 4 9 1 2 1 23 2 4 1 10 1 3 1 31 3 4 1 31	32
4 4 9 1 2 1 86 2 4 1 52 1 3 1 60 3 4 1 78	79
3 2 1000000000 3 2 1000000000 1 3 1 1 1000000000	1000000001
3 2 1 3 2 1000000000 1 3 1 1 1000000000	1000000001

Problem I. De Baikonur a Marte

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

O Cosmódromo de Baikonur, localizado no Cazaquistão, é a mais antiga instalação de lançamentos espaciais do mundo. Muitos voos históricos partiram de Baikonur: o primeiro satélite feito pelo homem, Sputnik 1, em 1957; o primeiro voo espacial tripulado por Yuri Gagarin, em 1961; e o voo da primeira mulher no espaço, Valentina Tereshkova, em 1963. Hoje, mais um foguete histórico está decolando, a humanidade está indo para Marte!

Você é um empregado alienígena da empresa Astronautas Chegando em Marte (ACM), e quer garantir que o foguete tenha espaço suficiente para um pouso seguro. Neste momento, o local onde os humanos vão pousar está ocupado por n montanhas, cada uma com altura a_i . Você deve destruí-las todas antes que os humanos cheguem.

Você pode realizar as seguintes duas operações:

- Selecionar um subconjunto S e diminuir a altura de todas as montanhas em S em uma unidade;
- Selecionar um subconjunto S e mudar a_i para $a_i \% 2$ (o resto da divisão de a_i por 2) para cada montanha em S .

Qual é o número mínimo de operações para fazer todas as montanhas terem altura zero?

Input

A primeira linha contém um número inteiro $1 \leq n \leq 10^5$. A segunda linha contém n inteiros a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$). É garantido que pelo menos uma montanha tem altura estritamente maior que zero.

Output

Imprima apenas um inteiro: o número mínimo de operações para fazer todas as montanhas terem altura zero.

Examples

standard input	standard output
5 1 1 1 1 1	1
4 9 0 8 2	2

Note

No primeiro caso, você seleciona todas as montanhas e diminui suas alturas em uma unidade. No segundo caso, você primeiro usa uma operação para diminuir a altura da primeira montanha e, em seguida, usa a operação do resto para destruir todas as montanhas.

Problem J. Acarajé

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

Moretev é um cazaque nascido em Astana que se mudou para Salvador e se apaixonou pelo Acarajé, grande expoente da culinária brasileira. Por causa disso, ele decidiu fundar a sua própria barraquinha de Acarajé e ao iniciar suas vendas se deparou com um enorme problema: ele é obrigado a definir o preço do produto antes de anunciá-lo.

“Que absurdo! Nos bazares da minha terra natal isso não acontece!”

Você foi contratado para ajudar Moretev a definir o preço do seu Acarajé a fim de maximizar os lucros. Para lhe ajudar, ele realizou uma pesquisa de campo e trouxe a seguinte amostra: uma lista de N consumidores, sendo que o consumidor i somente comprará o Acarajé se o mesmo estiver por um preço menor ou igual a p_i .

De acordo somente com essa lista, defina qual o preço que Moretev precisaria anunciar seu Acarajé e qual seria sua receita nessa amostra.

Input

A primeira linha contém um inteiro N ($1 \leq N \leq 10^6$), a quantidade de consumidores na lista. A segunda linha contém N inteiros p_i ($1 \leq p_i \leq 10^9$), representando a lista de acordo com o enunciado.

Output

Imprima uma linha contendo dois inteiros P e R , sendo P o preço que Moretev deve anunciar seu acarajé e R a receita que seria obtida. Se houver mais de um preço que maximize a receita, imprima qualquer um.

Examples

standard input	standard output
5 15 10 9 10 15	9 45
4 100 34 33 10	100 100
2 6 3	3 6

Problem K. Pegando pelúcias

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Memory limit: 1024 megabytes

Yan “o globe-trotter” Soares está viajando pela Ásia Central. Na cidade de Almaty, uma das maiores metrópoles do Cazaquistão, ele encontrou um curioso fliperama aberto 24 horas. Como ele gosta de jogos em geral, decidiu entrar nesta loja, e ali encontrou uma máquina de pegar pelúcias muito peculiar.

Dentro dessa máquina, as N pelúcias estão arranjadas em uma fileira. Além disso, a garra que é usada para pegar as pelúcias também é diferente. Essa garra tem duas unhas, esquerda e direita. A distância entre as unhas é exatamente a largura de W pelúcias (todas as pelúcias têm a mesma largura).

A seguir, descrevemos como a garra pega as pelúcias. Inicialmente, a garra está acima das pelúcias. O jogador escolhe a posição da unha esquerda (a unha direita está a distância W dela). A garra desce até que as unhas batem na fileira. Finalmente, a unha direita se move para a esquerda até que bate numa pelúcia (se entre as duas unhas há W pelúcias, a unha direita não se move). A garra retira todas as pelúcias que se encontram entre as duas unhas. O restante das pelúcias se mantém na mesma posição, ou seja, a fileira fica com posições vazias após usar a garra. Note que a unha esquerda não pode se mover para a esquerda da primeira pelúcia, e a unha direita não pode se mover à direita da última pelúcia da fileira.

Yan tem dado um valor de preferência p_i a cada pelúcia da fileira (esse valor pode ser negativo inclusive). Além disso, ele possui M moedas para jogar. Como não quer desperdiçar essas moedas, pediu a sua ajuda para determinar qual é o máximo valor total da preferência que ele pode atingir, se usar até M vezes a garra (ele pode não usar nenhuma moeda inclusive). Ajude ao Yan a encontrar esse valor máximo!

Input

A entrada é composta de duas linhas. A primeira linha contém três inteiros N ($2 \leq N \leq 10^4$), M ($1 \leq M \leq 5000$) e W ($1 \leq W \leq N$), o número de pelúcias, o número de moedas e a largura da garra, respectivamente. A segunda linha contém N inteiros que representam as preferências do Yan. O i -ésimo desses inteiros é o valor de preferência p_i ($-10^4 \leq p_i \leq 10^4$) que o Yan atribuiu à pelúcia i .

Output

Um inteiro que representa o máximo valor total da preferência que o Yan pode atingir usando até M moedas.

Example

standard input	standard output
4 2 3 1 8 -10 5	4

Problem L. Noite em Hazrat Sultan

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

Willian está viajando por Cazaquistão para descansar de seu trabalho e da faculdade, cansado de resolver inúmeros problemas de servidor na empresa. Durante a noite, Willian resolveu sair do hotel para aproveitar o lugar e apreciar a linda paisagem local, passando por inúmeros pontos turísticos como a torre Balterek, a mesquita de Hazrat Sultan, entre outros lugares. Ao olhar para o céu em frente à mesquita, Willian notou como o céu estava cheio de estrelas, acompanhado pelos minaretes iluminados da mesquita. As estrelas apareciam e sumiam no céu escuro, enquanto as nuvens na escuridão passavam sobre as estrelas. Nesse dia, Willian olhou para as estrelas e de tanto estudar geometria, fazer um monte de lista e exercício-programa, se perguntou:

— Qual é o par de estrelas que forma o maior valor absoluto de coeficiente angular se fosse uma reta?

Uma reta pode ser descrita algebricamente por equações lineares como $y = ax + b$, onde x é a variável independente da função $y = f(x)$, a constante a é o coeficiente angular e a constante b é o intercepto em y da reta. O valor absoluto é definido por uma função $y = |x|$, onde se $x < 0$, então $|x| = -x$ e se $x \geq 0$, então $|x| = x$.

Willian continuou apreciando a vista do céu e logo percebeu que o fenômeno das estrelas sendo coberto pelas nuvens pode ser representado da seguinte maneira. No início, é dada a configuração inicial de N estrelas. Depois disso temos Q eventos, onde os eventos são caracterizados por:

1. Uma estrela aparece em (X_i, Y_i) .
2. Uma estrela desaparece em (X_i, Y_i) .

Na configuração inicial das estrelas e após cada um dos eventos, queremos saber qual é o maior valor absoluto de coeficiente angular dos pares de estrelas.

Input

A entrada é dada por dois inteiros N e Q , seguido por N pares de inteiros representando a coordenada de cada estrela na configuração inicial. Após isso temos Q eventos descritos por 3 números inteiros, onde o primeiro valor define o tipo do evento, se for 1 é evento de inserção, se for 2 é evento de remoção e o segundo e terceiro valor definem a coordenada X_i e Y_i . Não existem duas estrelas na mesma linha e na mesma coluna, ou seja não existem duas estrelas com a mesma coordenada y e não existem duas estrelas com a mesma coordenada x . Na hora do desaparecimento da estrela é garantido que existe.

$$0 \leq N + Q \leq 2 \cdot 10^5$$

$$1 \leq X_i, Y_i \leq 10^9$$

Output

Imprima $Q + 1$ linhas com o maior valor absoluto de coeficiente angular. A primeira linha é a resposta para a configuração inicial e as Q linhas seguintes são a nova resposta após a execução de um evento.

Em cada um desses casos, imprima dois inteiros A e B que representem o maior valor absoluto de coeficiente angular como uma fração irredutível $\frac{A}{B}$, o que significa que $\text{mdc}(A, B) = 1$ (mdc é uma função que retorna o máximo divisor comum entre A e B). Se não existe nenhum par imprima -1 .

Examples

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