

Problem A. Troca de Bicicletas

Input file: standard input
Output file: standard output
Time limit: 2.5 seconds
Memory limit: 512 megabytes

Amy Rose faz parte de um grupo de N amigos e eles têm um costume estranho. Cada um possui uma bicicleta e, ao final de cada semana, o i -ésimo amigo dá sua bicicleta para o p_i -ésimo amigo (no caso de $i = p_i$, o amigo dá a sua bicicleta para si mesmo). É garantido que todos valores de p_i são distintos (P é uma permutação).



Amy, curiosa sobre o compartilhamento de bicicletas, quer saber qual o percurso que a bicicleta de cada amigo faz até voltar para ele, ou seja, a sequência de amigos que possuem a bicicleta do i -ésimo amigo, em ordem, até ela chegar nele mesmo novamente.

Input

A primeira linha da entrada contém um inteiro T ($1 \leq T \leq 100$) — o número de casos de teste. Daí seguem T casos de teste.

A primeira linha de cada caso de teste contém um inteiro N ($1 \leq N \leq 100$) — o número de amigos de Amy.

A segunda linha de cada caso de teste contém N inteiros p_1, p_2, \dots, p_n ($1 \leq p_i \leq N$, todos os valores de p_i são distintos entre si).

Output

Para cada caso de teste, imprima N linhas. A i -ésima linha deve conter a sequência, em ordem, de amigos que possuem a bicicleta que originalmente era do i -ésimo amigo, até que ela volte para ele.

É garantido que, eventualmente, para todos os casos de teste, a bicicleta retorna para seu proprietário original.

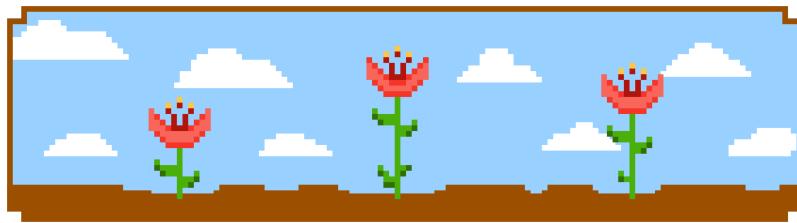
Example

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

Problem B. Cruzamento

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

Recentemente, enquanto se aventurava pelas florestas do planeta Möbius, Amy se deparou com uma rara espécie vegetal: a Rosita. Após muito tempo de estudo e observação, Amy conseguiu descrever o curioso processo reprodutivo dessa planta: todo organismo dessa espécie é hermafrodita, podendo cruzar com qualquer outro indivíduo para gerar uma planta filha. Além disso, quando uma Rosita de altura H_1 cruza com outra de altura H_2 , é gerada uma descendente de altura igual a média da altura dos pais: $(H_1 + H_2)/2$.



Ao fim de sua aventura, Amy recolheu N exemplares desta espécie para analisar cuidadosamente, e levantou a seguinte questão: dentre todas as formas de separar esses espécimes em $N/2$ pares e cruzar os indivíduos de cada par, qual minimiza a altura da maior planta filha?

Antes que pudesse sanar sua dúvida, as habilidades de Amy foram requisitadas em outra aventura. Agora, cabe a você ajudar Amy escrevendo um programa que responde à sua indagação.

Input

A primeira linha da entrada contém um inteiro N ($2 \leq N \leq 2 \cdot 10^5$, N é par), representando o número de espécimes analisados por Amy.

A segunda e última linha contém N inteiros H_i ($2 \leq H_i \leq 10^9$, H_i é par), representando a altura da i -ésima planta.

Output

Imprima um único inteiro H , representando a altura da maior planta filha em uma configuração ótima.

Note que H sempre é inteiro, pois as alturas das plantas H_i sempre é par. Dessa forma, você **não** deve imprimir casas decimais.

Além disso, note que cada Rosita deve pertencer a **exatamente um** dos $N/2$ pares.

Examples

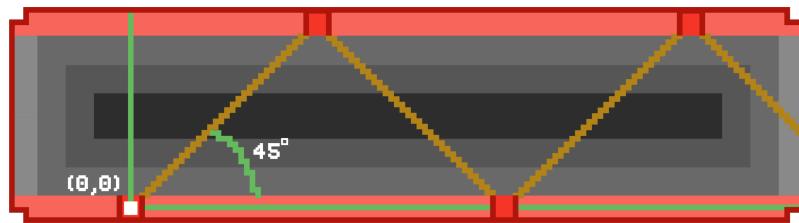
standard input	standard output
2 10 6	8
6 2 2 12 18 10 18	11

Problem C. Zigue-Zague

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

Na mais recente de suas aventuras, Amy se deparou com um árduo desafio: ela precisou correr sobre uma corda bamba para fugir do terrível Eggman.

A corda ligava dois prédios distintos de uma maneira curiosa: vistos de cima, os dois prédios pareciam segmentos de retas paralelos e a corda os ligava formando uma espécie de zigue-zague.



Considerando essa vista superficial no plano cartesiano, Amy percebeu que a corda começava na posição $(0, 0)$ e formava um ângulo de 45 graus sempre que alcançava um dos prédios.

Visando se preparar para desafios futuros, Amy precisa da sua ajuda para determinar até onde ela conseguiria correr nessas condições.

Mais especificamente, supondo Amy comece na posição $(0, 0)$, que a distância entre os prédios é D e que Amy consegue correr por M metros antes de se cansar, você deve dizer a posição (X, Y) em que ela vai parar.

Input

A única linha da entrada consiste dos dois inteiros D e M descritos no enunciado, separados por espaço. É garantido que $0 \leq M, D \leq 10^9$.

Output

Imprima dois números reais X e Y separados por espaço. Sua resposta será considerada correta se o erro absoluto ou relativo for menor ou igual a 10^{-6} .

Para imprimir números de ponto flutuante em python você pode usar a função round:

```
print(round(x,10),round(y,10))
```

Em C++ use o cout::setprecision:

```
cout << setprecision(10) << fixed;
cout << x << ' ' << y << endl;
```

Examples

standard input	standard output
1 2	1.4142135624 0.5857864376
10 5	3.5355339059 3.5355339059

Problem D. Constante Mística do Universo

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

Em todo universo existe uma constante mística que rege e coordena todos os eventos (por exemplo, no nosso universo esta constante é o 42). Em estudos recentes, Amy encontrou uma relação de forças entre moedas e acha que está muito perto de encontrar a constante mística do universo e do mundo para o universo Sonic! No entanto, existe uma superstição de que, se essa constante é negativa, o universo é condenado a falência, se for nula, o universo não evoluirá e se for positiva, o universo será próspero e feliz. A relação encontrada por Amy foi a seguinte:

$$F = \frac{G \cdot M_1 \cdot M_2}{D^2}$$

onde M_1 e M_2 são propriedades específicas das moedas e D é a distância entre elas. Nesse caso G é a constante mística do universo e do mundo e F é a força de atração entre as moedas.



Dados os valores de M_1 , M_2 , X_1 , X_2 e F , sendo: M_1 e M_2 as propriedades, X_1 e X_2 as posições das moedas e F a força entre elas, calcule G o valor de G para ajudar Amy a descobrir o futuro do universo.

Input

A primeira linha da entrada contém os valores de M_1 e M_2 .

A segunda linha contém os valores de X_1 e X_2 . É garantido que $X_1 \neq X_2$.

A terceira e última linha contém o valor de F .

É garantido que $-2^{31} \leq X_1, X_2, F \leq 2^{31} - 1$, $0 < M_1, M_2 \leq 2^{31} - 1$, e todos os valores são dados com duas casas decimais.

Output

Imprima em uma única linha o valor de G definido no enunciado. Sua resposta será considerada correta se o erro absoluto ou o erro relativo forem menores que 10^{-6} .

Para python você pode usar a função round:

`print(round(G, 10))`, para ter 10 casas, por exemplo.

Para C++, use:

`cout << setprecision(10) << fixed << G << endl;`

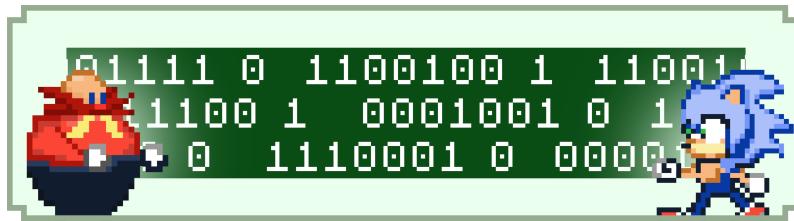
Examples

standard input	standard output
1.00 1.00 0.00 1.00 1.00	1.000000
1.00 1.00 0.01 2.01 1.00	4.000000
1.50 2.50 -1.30 1.30 4200.42	7571.957120

Problem E. Código

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

Amy Rose está em uma missão secreta para derrotar Eggman. Como parte de seu plano, ela teve que se separar de Sonic, e agora eles podem se comunicar apenas através de código Morse. Porém, como toda forma de comunicação, é possível que as mensagens sejam interpretadas de forma errada, gerando grande confusão.



Para lidar com isso, Amy desenvolveu um rebuscado esquema de segurança. Sabemos que as mensagens em Morse consistem de traços e pontos. Como amante da ciência da computação, Amy representa traços por um 1 e pontos por um 0, tendo a mensagem em binário. Foi combinado que toda mensagem teria 7 dígitos binários e por fim um bit de segurança. Esse bit é igual a 0 se a quantidade de 1's na mensagem for par e 1 caso contrário.

Dessa forma, se o bit de segurança for 0 e a mensagem tem um número ímpar de bits 1, então com certeza a mensagem foi corrompida. Por outro lado, mesmo que não haja uma contradição na versão criptografada, o bit de segurança não garante que a mensagem captada condiz com a original, uma vez que pode ocorrer de mais de 1 bit ter sido comprometido.

Dados os 8 bits (7 da mensagem mais o de segurança) sua tarefa é dizer se a mensagem certamente foi corrompida ou se não podemos concluir com certeza.

Input

A entrada é dada por 8 inteiros separados por espaço em uma única linha. É garantido que cada inteiro é ou 0 ou 1.

Output

Se você tem certeza que a mensagem foi corrompida, imprima apenas o caracter "S". Caso contrário, imprima "N?"(sem aspas).

Examples

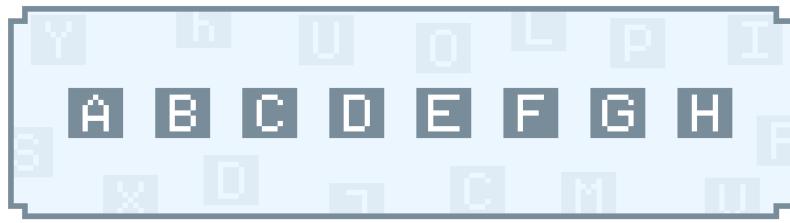
standard input	standard output
0 0 0 0 1 0 0 1	N?
0 1 1 1 1 0 1 1	N?
1 0 1 1 1 0 0 1	S
0 1 0 1 0 1 0 0	S

Problem F. Mensagem

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

"Não! Isso não é bom! Você estragou meu plano...!"

Eggman, cansado de ser batido em todo jogo, decidiu usar suas habilidades de cientista para interceptar mensagens entre universos distintos, a fim de descobrir novas tecnologias balísticas. Assim, ele conseguiu interceptar uma mensagem que ia do universo Hedgehg [sic] para o universo Sonicinos. Porém, apesar de seus 300 de QI, Eggman não conseguiu decifrar a mensagem e foi dormir, deixando-a dentro de uma caixa com metais valiosos.



Enquanto Eggman dormia, Rouge the Bat invadiu seu covil e roubou a caixa, sem esperar que dentro dela houvesse uma mensagem mais valiosa que os metais. Ao perceber isso, Rouge resolveu começar a decifrar a mensagem. Ela sabe que no universo Hedgehg existem apenas as letras de A até H, portanto a mensagem só contém essas letras. Além disso, ela sabe que no universo Sonicinos todas as palavras são palíndromos¹. Sabendo que a mensagem original não necessariamente é um palíndromo, Rouge resolveu olhar para cada subcadeia² da mensagem e verificar se é possível rearranjar seus caracteres de forma que a subcadeia se torne um palíndromo.

Porém, como a mensagem pode ser muito grande, essa tarefa pode ser muito trabalhosa. Por isso, Rouge pede a você que escreva um programa que receba a mensagem e conte quantas subcadeias dela podem ser rearranjadas de forma que se tornem palíndromos. As subcadeias não precisam ser distintas entre si.

¹: um palíndromo é uma cadeia de caracteres não vazia que se lê a mesma de trás para frente. Por exemplo, “ovo”, “arara” e “osso” são palíndromos, enquanto “uva” e “oi” não são.

²: uma subcadeia de uma mensagem é um segmento contínuo e não vazio de caracteres dessa mensagem. Por exemplo, as subcadeias de “amy” são “a”, “m”, “y”, “am”, “my” e “amy”.

Input

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 10^5$), o número de caracteres da mensagem. A segunda linha da entrada contém a mensagem S , uma palavra contendo N letras minúsculas de A a H.

Output

Imprima um inteiro, a quantidade de subcadeias da mensagem que podem ser rearranjadas como palíndromos.

Examples

standard input	standard output
3 egg	5
7 hedgehg	8
8 abcdefgh	8
7 abacaba	12

Note

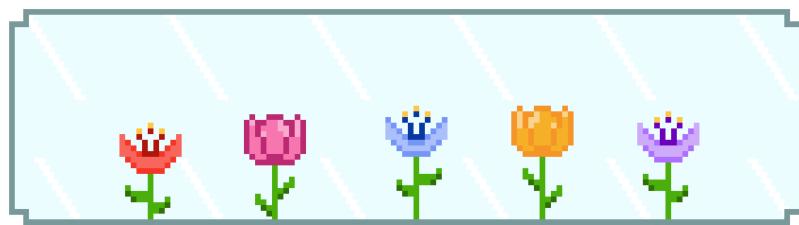
No primeiro exemplo, “egg” possui 5 subcadeias que podem ser rearranjadas como palíndromos: “e”, “g”, “g”, “gg” e “egg”. Note que existem duas subcadeias “g” porque são subcadeias de diferentes posições da palavra e as subcadeias não precisam ser distintas entre si. Note também que a subcadeia “egg” pode ser rearranjada como o palíndromo “gég”.

Problem G. Museu de flores

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

Fascinada pelo universo da botânica, Amy resolveu inaugurar o primeiro museu de flores do planeta Mobius. A principal atração do museu consiste em uma sequência de N flores (numeradas de 1 a N), na qual cada flor é uma Rosita ou uma Tulipa.

Como toda preparada gestora de negócios, Amy sabe que no dia da inauguração do museu receberá M convidados, e também sabe que o i -ésimo dos convidados visitará apenas o intervalo das flores de L_i até R_i (inclusivo).



Sendo assim, Amy definiu como a **satisfação** de um convidado o produto da quantidade de Rositas e de Tulipas que ele visita, e gostaria da sua ajuda para determinar que tipo de flor colocar em cada posição da sequência de forma a maximizar a soma da satisfação de todos os convidados.

Input

A primeira linha da entrada contém dois inteiros N e M ($1 \leq N, M \leq 10^3$), representando o número de flores na sequência da atração principal e a quantidade de visitantes no dia da inauguração do museu, respectivamente.

Cada uma das M linhas seguintes contém dois inteiros L_i e R_i ($1 \leq L_i \leq R_i \leq N$), representando o intervalo que o i -ésimo convidado visitará.

Output

Imprima uma sequência de N caracteres. O i -ésimo caracter deve ser «0» caso você queira colocar uma Rosita na i -ésima posição da sequência e «1» caso você queira colocar uma Tulipa.

Caso exista mais de uma sequência que maximiza a satisfação total dos visitantes, imprima qualquer uma.

Example

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

Problem H. Coleção de moedas

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

Amy, reconhecida em múltiplas galáxias por seus dons em organização, coleciona cuidadosamente todas as moedas que conquista em suas aventuras e as guarda sequencialmente em uma prateleira, na ordem em que as encontra. Como cada aventura traz lembranças diferentes para Amy, ela resolveu atribuir um valor numérico especial para cada moeda, chamado de **importância** da moeda, que representa o apreço de Amy por aquele item.

Infelizmente, Sonic, enquanto corria desatentamente mundo afora, acabou derrubando todas as moedas de Amy. Apesar de não se lembrar exatamente a ordem das moedas, Amy, curiosamente, se lembra da soma dos valores das importâncias para cada par de moedas adjacentes na ordenação original.



Agora, cabe a você, com suas habilidades excepcionais de programação, ajudar Amy a recuperar a ordem cronológica das moedas.

Input

A primeira linha da entrada contém um inteiro N ($2 \leq N \leq 5 \cdot 10^5$), representando o número de moedas da coleção de Amy.

A segunda linha contém N inteiros a_i ($1 \leq a_i \leq 10^9$), representando a importância da i -ésima moeda, em uma ordem qualquer deixada por Sonic.

A terceira e última linha contém $N - 1$ inteiros S_i ($2 \leq S_i \leq 2 \cdot 10^9$), representando a soma das importâncias das moedas adjacentes na ordem em que Amy se lembra. Mais formalmente, sendo $p = \{p_1, p_2, \dots, p_n\}$ a ordem original das importâncias, $S_i = p_i + p_{i+1}$, para $1 \leq i < n$.

Output

Imprima N inteiros p_1, p_2, \dots, p_n separados por espaço em uma única linha, representando a ordem original das importâncias das moedas.

Como Amy possui uma memória extraordinária, é garantido que existe uma solução para todos os casos de teste. Caso exista mais de uma solução, imprima qualquer uma.

Example

standard input	standard output
5 3 5 5 1 4 6 5 9 8	5 1 4 5 3

Problem I. Complicações logísticas

Input file: standard input
Output file: standard output
Time limit: 2.5 seconds
Memory limit: 256 megabytes

É fato que Amy já visitou muitos lugares estranhos em suas aventuras, mas um que se destaca é a cidade que ela escolheu para inaugurar o seu famoso museu de flores: a Linearlândia. Nesta singular cidade, todas as casas suportam no máximo um único morador e estão dispostas sequencialmente em uma linha, que pode ser representada pela reta real.



Um dos principais desafios administrativos que Amy tem enfrentado em sua jornada como empreendedora é decidir o local para construir o museu. Isso ocorre pois Linearlândia é uma cidade muitíssimo movimentada, sendo comum ocorrer de novos moradores chegarem à cidade e alguns se retirem. Por conta disso, ela precisa de sua ajuda para determinar a posição ideal para seu museu.

Para isso, você precisa escrever um programa que, dadas as informações sobre o trânsito de pessoas em Linearlândia, encontre a posição que minimiza a soma das distâncias do museu para cada um dos moradores.

Serão fornecidas Q atualizações sobre Linearlândia, sendo que cada uma pode ser de dois tipos distintos:

1. Um novo morador chegou à casa na posição X_i .
2. O morador da casa na posição X_i saiu de Linearlândia.

Você deve escrever um programa que, após cada atualização, diga qual a menor soma possível das distâncias do museu para cada uma das casas habitadas em uma disposição ótima.

Input

A primeira linha da entrada possui um único inteiro Q ($1 \leq Q \leq 2 \cdot 10^5$), a quantidade de atualizações sobre Linearlândia. Cada uma das próximas Q linhas possui um caractere T_i e um inteiro X_i ($1 \leq X_i \leq 10^9$), separados por espaço, representando o tipo da atualização e a posição correspondente.

Se T_i for "+", a i -ésima atualização corresponde à chegada de um novo morador na posição X_i .

Se T_i for "-", a atualização corresponde à saída de um morador da posição X_i .

Output

Imprima Q linhas, cada uma contendo um inteiro D_i , a menor soma possível após a i -ésima atualização, conforme descrito no enunciado. Caso nenhuma das casas esteja habitada após a i -ésima atualização, imprima -1.

Examples

standard input	standard output
5	0
+ 1	2
+ 3	2
+ 2	4
+ 4	6
+ 5	
10	0
+ 5	4
+ 1	4
+ 3	5
+ 2	6
+ 4	5
- 2	4
- 3	1
- 1	0
- 5	-1
- 4	

Problem J. Chaves e cadeados

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

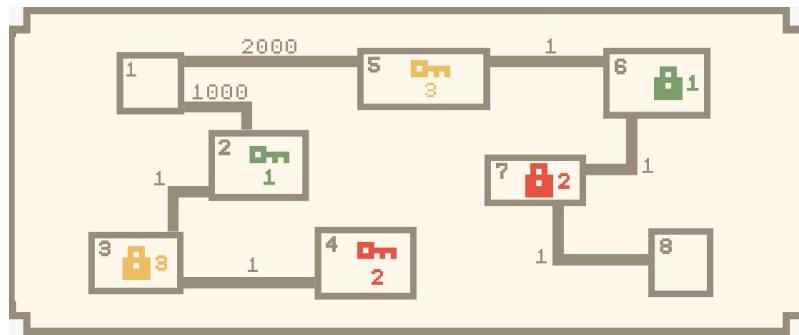
Amy Rose está cansada de brigar com Eggman constantemente, então decidiu propor-lhe uma trégua. Eggman aceitou mas somente sob uma condição: ele irá montar uma espécie de labirinto e caso Amy consiga escapar ele dará uma pausa em suas maldades.

O labirinto de Eggman consiste de N salas numeradas de 1 a N , das quais $N - 1$ pares estão conectados por túneis de distância variável. É garantido que é possível ir de qualquer sala a qualquer outra seguindo esses túneis. Todo túnel liga suas respectivas salas nos dois sentidos.

Amy foi colocada na sala 1 e a saída está na sala N . Nesse momento, Amy aceita o desafio de Eggman, pois acha que será moleza ir de uma sala a outra. Porém, para sua surpresa, Eggman coloca cadeados em algumas das salas e espalha as chaves. Se uma sala possui um cadeado, é impossível entrar nela por qualquer túnel a menos que esteja de posse da chave correspondente. Além disso, cada chave abre exatamente um cadeado.

Dado a estrutura do labirinto, isto é, quais os túneis e onde estão as chaves e cadeados, você deve ajudar Amy a completar esse desafio e lhe dizer qual o menor tempo que é preciso para escapar, ou afirmar que isto não é possível.

Note que você apenas gasta tempo para atravessar os túneis. O tempo de pegar chaves ou abrir cadeados é desprezível.



Representação do terceiro exemplo.

Input

A primeira linha da entrada possui dois inteiros N e K , $1 \leq N \leq 2 \cdot 10^5$ e $1 \leq K \leq 16$.

Na segunda linha temos as posições dos cadeados: $cad_1, cad_2, \dots, cad_k$.

Na terceira linha temos as posições das chaves no mesmo formato: c_1, c_2, \dots, c_k .

Note que a primeira chave (que está em c_1) abre o primeiro cadeado (em cad_1). De forma geral, você deve visitar c_i antes de cad_i para todo i . É garantido que não há cadeados ou chaves nem em 1 ou em N e que não há sala com um cadeado e uma chave ao mesmo tempo.

Por fim, seguem $N - 1$ linhas representando os túneis. A i -ésima dessas linhas tem 3 inteiros u_i, v_i e t_i representando um túnel bidirecional da sala u_i para v_i que tem tamanho t_i .

Output

Imprima um único inteiro -1 caso seja impossível completar o desafio. Caso contrário, imprima um único inteiro representando o menor tempo para completar o desafio.

Examples

standard input	standard output
4 1 2 3 1 2 1 2 3 1 3 4 1	-1
8 3 2 3 6 4 5 7 1 2 10 2 7 1 2 3 10 3 8 10 1 4 1000 4 5 1000 5 6 1	4030
8 3 6 7 3 2 4 5 1 2 1000 2 3 1 3 4 1 1 5 2000 5 6 1 6 7 1 7 8 1	8007