

## Problem A. Soma de ímpares

Input file:            standard input  
Output file:           standard output  
Time limit:           15 seconds  
Memory limit:         1024 megabytes

Quando criança, Techno escutou uma história quando Gauss também era criança e conseguiu calcular rapidamente a soma dos números de 1 até 100. A técnica de Gauss foi perceber que somar 1 com 100 dá o mesmo que 2 com 99, 3 com 98 ... até 50 com 51. Assim, a soma será 101 vezes o número de pares, que é 50.

$$\begin{array}{r} 1 + 2 + \dots + 50 \\ + 100 + 99 + \dots + 51 \\ \hline 101 + 101 + \dots + 101 \end{array}$$



Mini technoblade não impressionado pela ideia de Gauss escrita no quadro branco acima

Todos sempre se impressionam com essa história, menos Techno. Ele decidiu fazer algo mais difícil, e achar a soma dos primeiros  $n$  números ímpares:  $1 + 3 + 5 \dots + 2n - 1$ .

### Input

A entrada consiste de um único inteiro  $n$ ,  $1 \leq n \leq 10^7$ .

### Output

Imprima em uma única linha o inteiro representando a soma dos primeiros  $n$  números ímpares:  $1 + 3 + 5 \dots + (2n - 1)$ .

Note que esse número NÃO cabe em um inteiro de 32-bits. Em C/C++ use “long long int” para a resposta e para o  $n$ , ou então use python.

### Examples

standard input	standard output
1	1
2	4
3	9
10000000	100000000000000

## Note

Para  $n = 1$ , a soma é apenas 1.

Para  $n = 2$ , a soma é  $1 + 3 = 4$ .

Para  $n = 3$ , a soma é  $1 + 3 + 5 = 9$ .

Para  $n = 4$ , a soma é  $1 + 3 + 5 + 7 = 16$ . E assim por diante.

## Problem B. Potato War 1

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



*"Se você conhece o inimigo e conhece a si mesmo, não precisa temer o resultado de cem batalhas. Sun Tzu, A arte da guerra"*

Technoblade é famoso por sempre levar as coisas ao extremo. Um dia ele estava plantando batatas e, após ver o tamanho da sua enorme plantação, ficou curioso para saber quanto tempo demoraria para ele se tornar a pessoa com mais batatas. Após investigação, Techno fez as contas e viu que ele tem  $X_1$  batatas e produz  $Y_1$  batatas por dia, enquanto o seu maior concorrente, Squid Kid, tem  $X_2$  batatas e produz  $Y_2$  batatas por dia.

Ele se declarará vitorioso da Primeira Guerra das Batatas se tiver um número **estritamente** maior de batatas que Squid. Ajude Techno a prever o resultado dessa batalha.

### Input

O input é apenas uma linha com 4 inteiros  $X_1, Y_1, X_2, Y_2$  separados por espaço, nessa ordem. É garantido que todos os números são maiores ou iguais a 0 e menores ou iguais a 1000.

### Output

Caso Technoblade esteja destinado ao sucesso e a vitória na Primeira Guerra das Batatas, imprima a menor quantidade de dias até que ele tenha estritamente mais batatas que SquidKid. Caso ele ainda não tenha uma plantação grande o suficiente, e não irá ganhar a batalha em um tempo finito, imprima  $-1$ .

Vale notar que a quantidade de dias deve ser um número inteiro, pois só no final do dia as batatas são contabilizadas.

### Examples

standard input	standard output
1000 0 999 1000	0
0 1 1000 0	1001
100 2 200 1	101

## Problem C. Vendas

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1 second  
Memory limit:         1024 megabytes

Para uma empresa saber se ela está indo bem ou mal nas vendas é importante analisar a média das vendas em um intervalo de tempo.

Como analista de vendas, você observa intervalos de  $K$  meses consecutivos e encontra a média. Você está interessado no primeiro intervalo com a menor média e no primeiro com a maior média.

Formalmente, podemos representar as vendas como uma lista  $v_1, v_2 \dots v_N$  de  $N$  meses, onde  $v_i$  quer dizer quantos itens foram vendidos no mês  $i$ . Um intervalo que começa no mês  $i$  e tem tamanho  $K$  engloba os elementos  $v_i, v_{i+1} \dots v_{i+K-1}$  e tem média  $\frac{(v_i + v_{i+1} \dots + v_{i+K-1})}{K}$ .

Por exemplo se  $v = \{1, 1, 1, 2, 3\}$  e  $K = 2$  o primeiro intervalo com menor média começa em  $i = 1$ ,  $\frac{(v_1 + v_2)}{2} = \frac{(1+1)}{2} = 1$ , e o com maior média começa em  $i = 4$ ,  $\frac{(v_4 + v_5)}{2} = \frac{(2+3)}{2} = 2.5$ . Note que o intervalo começando em  $i = 2$  também tem média 1, mas não é o primeiro intervalo com menor média. Note também que  $i = 5$  não delimita um intervalo válido, pois ainda não sabemos o número de vendas no mês 6.

### Input

A primeira linha contém dois inteiros  $N, K$ , o número de meses da lista de vendas e o tamanho do intervalo. É garantido que  $1 \leq K \leq N \leq 10^5$ .

Na segunda linha há  $N$  inteiros  $v_1, v_2 \dots v_N$ , representando a lista de vendas. É garantido que  $1 \leq v_i \leq 10^4$ .

### Output

Imprima uma linha com dois inteiros: o índice do começo do primeiro intervalo com a menor e com a maior média.

Note que o valor máximo do  $N$  é grande. Se estiver recebendo TLE excedido pode ser útil pensar em formas mais otimizadas de se fazer esse algoritmo.

### Examples

standard input	standard output
5 2 1 1 1 2 3	1 4
6 4 1000 1 2 3 4 100	2 1

## Problem D. Skywars

Input file:           standard input  
Output file:         standard output  
Time limit:          0.5 seconds  
Memory limit:       256 megabytes



Um mapa de Skywars

Skywars é um minigame em que cada jogador começa em uma ilha voadora e o objetivo é ser o último sobrevivente. Techno está jogando uma partida, e agora só resta ele e 2 inimigos. Uma visão de cima das ilhas mostra pontos vazios e pontos em que existem blocos de ilhas, de tal forma que é possível representar o mapa como um grid. Para ganhar, Techno deve chegar nos seus inimigos. Por isso, ele quer saber qual a menor quantidade de blocos que devem ser colocados de modo que a sua ilha e a dos 2 inimigos estejam conectadas. Aqui 2 blocos são considerados conectados se é possível ir de uma a outra se movendo apenas para as direções adjacentes no grid: cima, baixo, esquerda e direita. Note que diagonais não contam.

### Input

A primeira linha contém 2 inteiros,  $N$  e  $M$ , representando o número de linhas e o número de colunas do grid, respectivamente. É garantido que  $1 \leq N, M \leq 2 \cdot 10^5$  e que  $N \cdot M \leq 2 \cdot 10^5$ .

Seguem  $N$  linhas com  $M$  caracteres  $c_i$  cada. É garantido que  $c_i \in \{T, *, ., \#\}$ . 'T' representa onde Techno está e o asterisco ('\*') o lugar dos inimigos. É garantido que há exatamente um 'T' e exatamente 2 asteriscos. O caractere "ponto"('.') representa células sem blocos enquanto hashtag ('#') células com blocos.

### Output

Imprima um único inteiro representando o menor número de blocos necessários para conectar a ilha de Technoblade a de seus inimigos.

## Examples

standard input	standard output
4 4 T..* .... .... *...	4
4 4 *... .T.. ..*. ....	2
4 4 T#.* ..#. #..# ##.*	2
3 3 .*. T.* ...	1

## Note

Note que as células com Techno ou seus inimigos já possuem blocos.

## Problem E. Potato War 2

Input file: standard input  
Output file: standard output  
Time limit: 0.8 seconds  
Memory limit: 512 megabytes



*"Sometimes it's tough being the best..."*

Após quintuplicar o tamanho da sua plantação de batatas, agora Technoblade ficou entendiado de plantar batatas e decidiu comprar batatas, por motivos desconhecidos já que ele já tinha muitas batatas...

Ele fez uma lista de  $N$  lojas de batatas e viu que a  $i$ -ésima loja vende em pacotes de  $b_i$  batatas por pacote. Techno quer saber de quantos modos ele consegue comprar exatamente  $B$  batatas.

Como ele suspeita que a loja número 1 é do seu rival SquidKid, ele quer comprar no máximo  $t$  pacotes da loja 1. Ele pode comprar qualquer quantidade de pacotes das outras lojas, inclusive 0.

De quantos modos ele pode fazer essa compra? Dois jeitos são considerados diferentes se o número de pacotes de uma loja comprados em uma encomenda é diferente da outra. Como esse número é grande, imprima a resposta módulo  $10^9 + 7$ .

### Input

A primeira linha contém dois inteiros  $N$  e  $B$ ,  $1 \leq N \leq 100$  e  $1 \leq B \leq 10^{18}$ .

A segunda linha contém  $N$  inteiros  $1 \leq b_i \leq 500$ . Também é garantido que  $\sum_{i=1}^N b_i \leq 500$ .

Por fim, segue uma linha com o valor  $0 \leq t \leq 10^{18}$ .

## Output

Imprima uma única linha com um inteiro  $X$ ,  $0 \leq X < 10^9 + 7$ , o número de modos de comprar exatamente  $B$  batatas.

## Examples

standard input	standard output
1 10 1 9	0
2 3 1 1 1000	4
2 3 1 1 0	1
3 1000000000 1 2 3 1000000000	2500000003



## Problem F. Férias

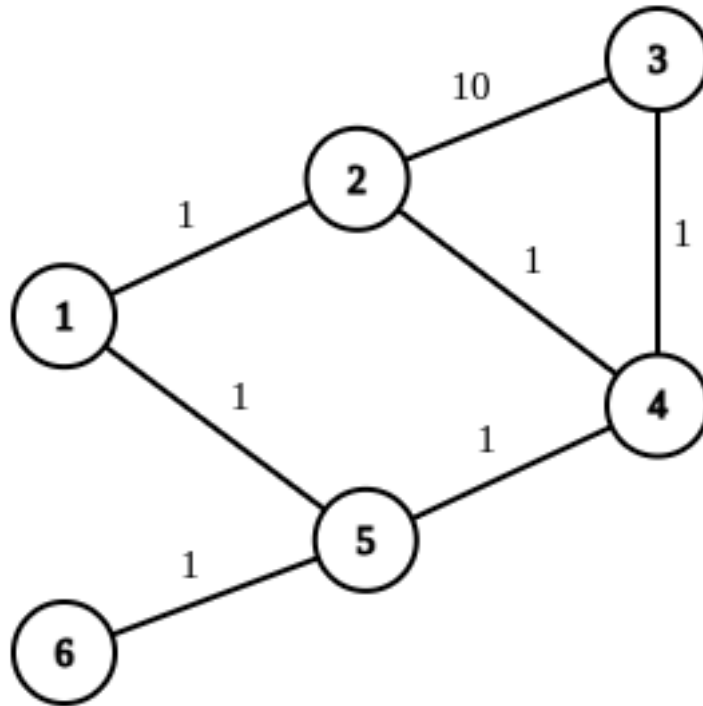
Input file:           standard input  
Output file:         standard output  
Time limit:          1.2 seconds  
Memory limit:       512 megabytes



Technoblade finalmente decidiu tirar férias. Ele decidiu que vai visitar uma lista extensa de cidades e para isso te contratou para fazer um plano de viagem. Para isso, toda cidade recebeu um identificador: um número inteiro de 1 a  $N$ , sendo 1 a cidade natal de Techno. Além disso, você anotou todas as viagens disponíveis, que podem ser representadas por 3 inteiros  $a_i, b_i, c_i$ , representa uma rota de  $a_i$  para  $b_i$  que custa  $c_i$  reais e pode ser realizada nas **duas direções**, de  $a_i$  para  $b_i$  e de  $b_i$  para  $a_i$ .

Agora resta você decidir em qual ordem visitar as cidades. Esse plano deve começar na cidade natal (número 1) e terminar nela. Além disso, toda cidade deve ser visitada **pelo menos** uma vez, sendo permitido repetições.

Como Technoblade está de férias ele não está tão preocupado com o dinheiro, e qualquer plano que tenha custo menor ou igual a 2 vezes o custo de um plano ótimo será aceita. Por exemplo, se tratarmos esse problema como um problema de grafos podemos ter um caso como o abaixo:



Cidades são representadas por vértices e as rotas por arestas entre os vértices. Os inteiros nas arestas representam seus custos

Note que o plano de menor custo tem custo 8.  $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 5 \rightarrow 1$  é um exemplo de plano.

Porém, o plano  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 5 \rightarrow 1$  tem um custo de  $16 \leq 2 \cdot 8$  então também é um plano válido.

## Input

A primeira linha contém 2 inteiros  $1 \leq N \leq 3 \cdot 10^5$  e  $1 \leq M \leq 5 \cdot 10^5$ , representando o número de cidades e o número de rotas. Seguem  $M$  linhas com 3 inteiros  $a_i, b_i, c_i$  cada. É garantido que  $1 \leq a_i, b_i \leq N$ ,  $a_i \neq b_i$  e  $1 \leq c_i \leq 10^{12}$ . Também é garantido que é possível fazer um plano que visite toda cidade apenas usando as rotas dadas.

## Output

Imprima na 1 linha o custo do seu plano encontrado. Na segunda linha imprima um inteiro  $K$  representando o número de cidades no seu plano, incluindo as repetições de uma mesma cidade. Por fim, imprima  $K$  inteiros  $1 \leq p_i \leq N$  representando as cidades a serem visitadas na ordem do seu plano. Você deve garantir que  $p_1 = p_K = 1$  e que para todo  $1 \leq i < K$  existe uma rota de  $p_i$  para  $p_{i+1}$ . Além disso, você deve garantir que  $K \leq 2 \cdot 10^6$ . É garantido que existe uma resposta que satisfaz essas condições.

## Examples

standard input	standard output
2 2 1 2 1 1 2 10	2 3 1 2 1
3 3 1 2 100000000000 1 3 100000000000 2 3 100000000000	400000000000 5 1 2 1 3 1

## Problem G. Beautiful Crown

Input file:           standard input  
Output file:         standard output  
Time limit:          3 seconds  
Memory limit:       1024 megabytes



Technoblade também é famoso por ter uma coroa bonita. Agora ele quer fazer novas coroas, uma com  $K$  espaços para joias, onde ele pode colocar qualquer um dos  $M$  tipos de joias conhecidos por ele.

Techno ficou surpreso pelo número de opções e agora está curioso em saber quantas coroas distintas de tamanho  $K$ , usando  $M$  tipos de joias existem. Todas as joias do mesmo tipo são idênticas, e duas joias de tipos diferentes sempre são distinguíveis. Note também que se é possível rotacionar uma coroa de modo que ela fique igual a outra, elas são consideradas idênticas. Por fim, todos os  $K$  espaços devem estar ocupados por alguma joia.

Como Techno não decidiu o valor de  $K$ , ele está interessado no somatório do número de coroas para todo  $K$  de 1 a  $N$ . Como esse número é grande, imprima módulo  $10^9 + 7$ .

### Input

A entrada são apenas dos inteiros  $N$  e  $M$ ,  $1 \leq N, M \leq 10^6$ .

### Output

Imprima uma única linha contendo o número de possíveis coroas com tamanho de 1 a  $N$ , usando  $M$  tipos de joias, módulo  $10^9 + 7$ .

## Examples

standard input	standard output
1 2	2
2 2	5
10 3	9503
1000 1000000	250445517

## Problem H. Divisão de Times

Input file:           standard input  
Output file:         standard output  
Time limit:          4.5 seconds  
Memory limit:       256 megabytes



Technoblade está organizando um jogo de Hunger Games, um modo de jogo do tipo “último sobrevivente vence”. Antes do início, os jogadores são colocados em um círculo. Então, eles são divididos em times. Para facilitar a organização, os times devem ser de jogadores adjacentes no círculo.

Technoblade já colocou os jogadores em um círculo e agora basta dividir os times. Os jogadores foram numerados de 1 a  $n$ , com o jogador  $i$  do lado do  $i + 1$  para  $i < n$  e o jogador 1 do lado do jogador  $n$ . Ele sabe o país de origem de cada jogador  $p_i$  e para diversificar, quer que todo time tenha no máximo  $k$  jogadores do mesmo país. Lembre-se que os jogadores do mesmo time devem estar um do lado do outro. Por exemplo, se  $n = 5$  podemos ter um time com os jogadores 1, 5 e no outro 2, 3, 4, pois 5 está do lado de 1 e 2, 3, 4 estão em sequência. Porém o time 1, 3 seria proibido. Vale ressaltar que cada jogador deve estar em **exatamente** um time. Para que a partida seja rápida, você quer minimizar o número de times, satisfazendo que tem no máximo  $k$  pessoas de cada país em cada time.

Como Techno ainda não decidiu o valor de  $k$ , ache a resposta para todo  $k$  de 1 a  $n$ .

### Input

A primeira linha contém um único inteiro  $n$ ,  $1 \leq n \leq 10^5$ .

Segue uma linha com  $n$  inteiros  $1 \leq p_i \leq n$  como descrito.

### Output

Imprima  $n$  linhas. A  $k$ -ésima sendo a menor quantidade de times para que todo time tenha no máximo  $k$  pessoas do mesmo país.

## Examples

standard input	standard output
5 1 2 1 2 1	3 2 1 1 1
6 1 2 2 1 2 1	3 2 1 1 1 1

## Note

Para usuários de C++: cuidado com o 'endl' para imprimir saídas grandes.

## Problem I. Username

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



Minecraft usernames

Tiago Domingos quer criar uma conta no Minecraft, para jogar junto com seu amigo Tiago Gonçalves. Para a escolha do nome de usuário, ele quer ser o mais original possível.

O username de Domingos pode ser apresentado por uma string  $D$  e o de Gonçalves uma string  $G$ . Agora, para medir quão original Domingos foi, ele acha a menor substring de  $D$  que não é substring de  $G$ . Sua tarefa é imprimir o tamanho da menor substring ou avisar caso todas as substrings de  $D$  sejam substrings de  $G$  também.

Por exemplo, se o username de Tiago Domingos for  $D = tdas$  e o de Gonçalves  $G = tfg$ , a resposta é 1, pois  $a, d, s$  não ocorrem em  $G$ .

Se  $D = tdas$  e  $G = tdas$  não há substring que não aparece em  $G$ .

Se  $D = td$  e  $G = tudo$ , a menor substring é  $td$  de tamanho 2.

### Input

A primeira linha tem a string  $D$  e a segunda  $G$ . É garantido que as strings contêm apenas letras latinas minúsculas ('a' até 'z') e que o tamanho delas é no máximo  $10^5$ .

### Output

O tamanho da menor substring de  $D$  que não é substring de  $G$ . Imprima  $-1$  se toda substring de  $D$  é substring de  $G$ .

## Examples

standard input	standard output
aba abacaba	-1
aaba aabbaa	3
tdas tdas	-1
td tudo	2

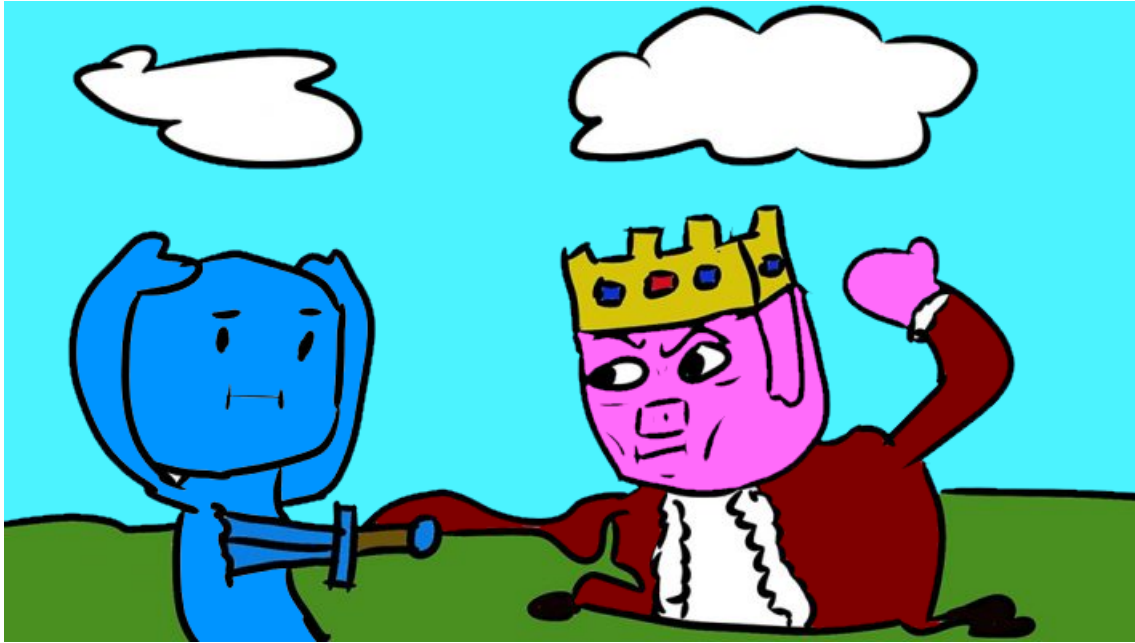
## Note

A substring é uma cadeia contínua de caracteres de uma string. Por exemplo, as substrings de *abc* são: *a*, *b*, *c*, *ab*, *bc* e *abc*. Note que *ac* **não** é substring de *abc*.



## Problem J. The Final Reckoning

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



*"If you wish to defeat me, train for another 100 years!"*

Technoblade e Skeppy decidiram jogar um jogo com moedas para passar o tempo. Cada um deles trouxe  $n$  moedas de casa e agora eles colocaram todas as  $2 \cdot n$  moedas em uma linha. Toda moeda tem um valor que pode ir de 1 real a  $10^4$  reais (não me pergunte como).

O jogo acontece em turnos, nos quais o jogador deve pegar exatamente uma moeda entre a mais a esquerda e a mais a direita para si. No final, a sua pontuação é o valor da soma das suas moedas. Quem tiver mais é considerado o campeão.

Como Technoblade é o mais velho, ele começa com 1 **real de vantagem**, mas em troca deve ser o primeiro a jogar.

Skeppy não gosta de pensar em estratégias, então ele apenas pega a maior entre a moeda da esquerda e da direita. Se as duas tem o mesmo valor ele prefere a da direita.

Você como espectador quer tentar prever o jogo. Você deve adivinhar quem irá ganhar o jogo, ou se Techno consegue forçar um empate, assumindo que Technoblade, diferente de Skeppy, pode jogar de forma ótima.

Caso Technoblade seja o campeão, imprima também a sua sequência de jogadas. Em caso de que muitas sequências garantam a vitória, imprima a com menor valor lexicográfico (veja output).

### Input

A primeira linha contém um inteiro  $n$ . Segue uma linha com  $2 \cdot n$  inteiros de 1 a 10000, representando as moedas da esquerda para a direita, já prontas para o início do jogo. É garantido que  $1 \leq n \leq 2500$ .

### Output

Caso Technoblade esteja fadado a derrota imprima apenas “:”, sem aspas.

Caso Technoblade não consiga ganhar, mas consiga forçar um empate, imprima apenas “tie”, sem aspas.

Caso ele consiga ganhar, imprima na primeira linha “TECHNOBLADE NEVER DIES!”, sem aspas. Imprima a seguir outra linha com  $n$  caracteres ‘L’ ou ‘R’. O  $i$ -ésimo caractere determina se Techno deve retirar a moeda da esquerda ou da direita nessa rodada. Se houver muitas sequências possíveis, imprima a menor lexicograficamente.

## Examples

standard input	standard output
2 1 10 10 1	TECHNOBLADE NEVER DIES! LL
3 10 1000 1 1 1 1	TECHNOBLADE NEVER DIES! RLL
2 1 10 1 10	TECHNOBLADE NEVER DIES! LL

## Note

Ordem lexicografica também é popularmente conhecida como ordem do dicionário.

No primeiro exemplo,  $RR$  também é uma sequência válida para ganhar o jogo, mas viria depois que  $LL$  no dicionário, portanto não é a resposta.

## Problem K. Otimismo

Input file: standard input  
Output file: standard output  
Time limit: 5 seconds  
Memory limit: 1024 megabytes



*"It's all part of my master plan"*

Agora que Technoblade já construiu seu império de batatas e tirou férias para descansar, ele decidiu apenas observar o desempenho da sua plantação. Techno tem  $n$  lotes, numerados de 1 a  $n$ . Para todo lote, seu terreno vale um preço que depende do clima, tipo da terra, erosão do solo, nível de chuva, etc.

Para se manter atualizado, ele primeiro fez uma lista dos valores dos terrenos:  $v_1, v_2, \dots, v_n$ , com  $v_i$  o preço inicial do  $i$ -ésimo terreno. Após isso, ele observou que os efeitos climáticos afetam um segmento contínuo dos lotes de cada vez, adicionando ou subtraindo no valor do lote. Por isso você deve responder a dois tipos de operação:

- Tipo 1: essa é a operação de adição no valor dos lotes. Ela tem o formato  $1 \ l \ r \ x$ . Com essa operação,  $v_i := v_i + x$  para todo  $l \leq i \leq r$ . É garantido que  $-10^6 \leq x \leq 10^6$ .
- Tipo 2: Tem o formato  $2 \ l \ r$ . Essa é uma operação de consulta, em que você deve imprimir a soma dos valores **otimistas** dos lotes de índices  $l$  a  $r$ . O valor otimista do lote é o maior valor que ele já teve até agora.

Por exemplo, se  $v = \{1, 2, 3\}$  e acontece uma operação do tipo  $1 \ 1 \ 2 \ 2 \ -2$ , ou seja,  $l = 2, r = 2$  somando  $x = -2$  temos que  $v = \{1, 0, 3\}$ . Além disso,  $v_{\text{otimista}} = \{1, 2, 3\}$  pois o maior valor que  $v_2$  teve foi 2.

### Input

A primeira linha tem dois inteiros  $n$  e  $q$ , representando o número de lotes e o número de operações. É garantido que  $1 \leq n, q \leq 3 \cdot 10^5$ .

Segue uma linha com  $n$  inteiros  $v_i$ ,  $1 \leq v_i \leq 10^6$ .

Por fim, seguem  $q$  linhas com as operações do tipo 1 ou tipo 2. As operações do tipo 1 tem o formato  $1 \ l \ r \ x$  e as do tipo 2 tem formato  $2 \ l \ r$ . É garantido que  $1 \leq l \leq r \leq n$  e  $-10^6 \leq x \leq 10^6$ .

## Output

Para toda pergunta do tipo 2 imprima o valor da soma dos valores otimistas dos lotes de número de  $l$  a  $r$ .

## Examples

standard input	standard output
1 5 1 1 1 1 99 1 1 1 -99 2 1 1 1 1 1 100 2 1 1	100 101
4 8 1 2 3 4 2 1 4 1 1 4 -1 2 1 4 1 1 3 2 2 3 3 1 1 3 5 1 1 3 -100 2 3 3	10 10 4 9

## Note

Note que valor  $v_i$  pode ficar negativo, o que quer dizer que a terra está tão acabada que ele pagaria pra o comprador.

## Problem L. Experimento de F129

Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         512 megabytes



Steve relaxando no seu bloco flutuante

Steve está fazendo um experimento de física para a matéria de *F129* da UNICAMP. Ele mediu quanto tempo demora para ele cair de cima de uma árvore e depois de somada as incertezas tem um intervalo de confiança  $[l, r]$  para a constante de gravidade. Depois ele refez o experimento jogando terra, madeira, tijolos e o que quer que encontrasse no caminho. No final, ele obteve  $n$  possíveis intervalos  $[l_1, r_1], [l_2, r_2] \dots [l_n, r_n]$ . Porém ele notou que alguns intervalos não tem interseção, o que seria péssimo para seu relatório. Sendo assim, ao invés de refazer os experimentos e entender o que fez de errado, ele optou por pegar a maior quantidade de intervalos que tenham interseção não vazia e rezar para que o valor correto da constante da gravidade esteja ali.

Formalmente, você deve escolher o maior número  $k$  de intervalos  $i_1, i_2, \dots, i_k$  tal que para quaisquer dois intervalos  $i_a$  e  $i_b$ , eles tenham interseção. Interseção nas pontas é válida, pois os intervalos são fechados.

### Input

Na primeira linha da entrada segue apenas um inteiro  $1 \leq n \leq 2 \cdot 10^5$ . Seguem  $n$  linhas com dois inteiros cada,  $l_i$  e  $r_i$ , representando os intervalos. É garantido que  $1 \leq l_i \leq r_i \leq 10^9$ . É possível que dois intervalos sejam iguais.

### Output

Na primeira linha imprima um inteiro  $k$ , representando o maior número de intervalos com interseção comum. Em seguida imprima  $k$  inteiros distintos de 1 a  $n$ , representando os índices dos intervalos usados. Se existirem muitas respostas com  $k$  máximo, qualquer uma é considerada correta.

## Examples

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