

Blasphemous

Autor: Manollo Castro de Pinho Martinez



Blasphemous é um jogo no estilo Roguelike que vai ficando cada vez mais desafiador de acordo com seu progresso dentro do jogo. Você se depara com muitos chefões que irão testar sua paciência para poder passar por eles. Além disso, nesse jogo temos também os altares que são chamados de “*Prie Dieu*” e neles você recarrega a sua vida e salva o game.

Tarefa

Lucas Souza está viciado no jogo, então ele decidiu simular o dano máximo que ele leva de todos os chefes, além de verificar se os altares de Prie Dieu são suficientes para salvá-lo. Assim, ele pediu pra você desenvolver um programa que dadas as informações sobre as fases do jogo, tais como altares e o dano dos chefões, determine se ele conseguiria ou não terminar o jogo.

Entrada

Na primeira linha da entrada será dado um inteiro ‘**N**’ ($1 \leq N \leq 100$), que indica o número de fases do jogo. Na linha seguinte serão dados ‘**N**’ inteiros ‘**X**’ ($0 \leq X \leq 1000$) que representam cada fase do jogo, sendo que ‘**X**’ pode ter três valores diferentes: I) fase com caminho livre é representada pelo valor 0 (zero); II) fases com altares de Prie Dieu serão identificadas com valor 1 (um); III) e as fases que tiverem chefes serão identificadas com valores inteiros no intervalo $2 \leq X \leq 1000$, **sendo que este valor inteiro é o dano que o chefe causa ao personagem naquela fase**. Por fim será dado um inteiro ‘**M**’ ($1 \leq M \leq 1000$), que representa a vida do Penitente.

Saída

A Saída vai consistir de uma frase. Caso Lucas consiga passar de todos os chefões ele deve imprimir “**Yes, you can**”, caso contrário deve imprimir “**You Died**”.

Entrada	Saída
12 0 0 1 0 0 99 0 0 1 85 0 14 100	Yes, you can
10 0 1 0 0 0 32 0 31 1 0 60	You Died

Dr. Strange e as Multidimensões

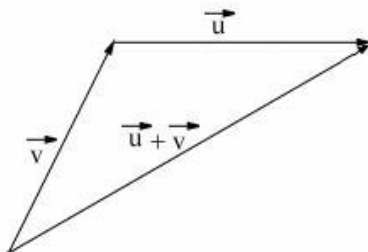
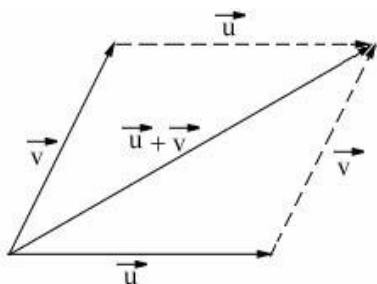


Não é novidade para ninguém que o **Dr. Strange**, o famoso mago da Marvel, tem a incrível habilidade de viajar entre dimensões. Mas poucos sabem que realizar tal feito não é tão fácil quanto parece.

Primeiro, Dr. Strange precisa pensar na dimensão que ele quer chegar. Para isso, ele pensa em um **número inteiro** ' N ' ≥ 2 , indicando o número da dimensão de destino. Por exemplo, se Dr. Strange deseja viajar para a sexta dimensão, ' $N = 6$ ', significa que cada ponto dessa dimensão pode ser especificado por uma **sequência** de **seis** números $S = (n_1, n_2, n_3, n_4, n_5, n_6)$. Assim, se Dr. Strange deseja viajar para a ' N -ésima dimensão, cada ponto dessa dimensão pode ser especificado por uma sequência de ' N ' números $S = (n_1, n_2, \dots, n_n)$, onde n_i é um número

real qualquer, para $1 \leq i \leq n$.

O próximo passo é mais complicado. Escolhida a dimensão, Dr. Strange escolhe o ponto dessa dimensão para onde ele deseja viajar, denotado D . A questão é que Dr. Strange não pode viajar diretamente para D , primeiro ele deve viajar para um ponto intermediário, e de lá viajar para D . Essa operação de viajar para um ponto e depois para o seu destino pode ser especificada como uma **soma de sequências** na mesma dimensão, ou seja, dado o destino D e duas sequências ' u ' e ' v ', todos na dimensão ' N ', sua tarefa é descobrir se a soma das duas sequências irá levar Dr. Strange para o ponto D . A soma de duas sequências é definida da seguinte forma: seja $u = (a_1, a_2, \dots, a_n)$ e $v = (b_1, b_2, \dots, b_n)$, $u + v = (a_1 + b_1, a_2 + b_2, \dots, a_n + b_n)$.



Entrada:

A primeira linha da entrada contém um inteiro ' N ' ($1 \leq N \leq 1000$). Indicando a dimensão para a qual Dr. Strange deseja viajar. As próximas três linhas indicam, respectivamente, os ' N ' números inteiros de cada uma das sequências u , v , e do ponto D .

Saída:

A saída é composta de uma única linha com as possíveis respostas sem aspas: "**OK**", se Dr. Strange consegue viajar para D , dadas as sequências u e v , e "**NOPE :(**", caso contrário.

Entrada	Saída
4 1 2 3 4 4 7 9 1 5 9 12 5	OK
3 1 2 4 1 56 22 1 2 4	NOPE :(

Em Busca da Esmeralda

Sonic perdeu uma das Esmeraldas do Caos!! Se essa jóia cair nas mãos do Dr. Eggman ninguém sabe do que ele seria capaz. Sonic sabe que a esmeralda está dentro de uma caixa que tem um determinado número. Sua tarefa é, dada uma sequência de caixas e o número da Esmeralda do Caos, determinar se Sonic pode recuperar a Esmeralda.



Entrada

A entrada possui três linhas. A primeira linha contém um inteiro **'N'** ($1 \leq N \leq 1000$), representando o número de caixas onde pode estar a Esmeralda do Caos. A segunda linha contém uma sequência de **'N'** inteiros n_1, n_2, \dots, n_n representando os números das esmeraldas que estão nas caixas, sendo $1 \leq n_i \leq 1000$. Por fim, na terceira linha será dado um inteiro **'C'** ($1 \leq C \leq 1000$) que é o número da Esmeralda do Caos.

Saída

A saída consiste em uma única linha contendo um número inteiro: O número da Esmeralda do Caos, caso ela se encontre na sequência de caixas dada, ou -1 caso contrário.

Entrada	Saída
8 1 2 4 3 5 6 7 8 4	4
10 12 1 4 7 9 8 1 2 3 90 6	-1

O mochileiro metódico



Will é um vendedor que mora na vila de Rynoka e sai em aventuras desbravando masmorras para conseguir bons equipamentos e vendê-los em sua loja. Mas Will é um rapaz muito metódico, e, toda vez que retorna de uma jornada, ele esvazia a sua mochila retirando um item por vez, sempre do último adquirido até o primeiro. Sua tarefa é fazer um programa que, dado uma lista de itens na mochila de Will, você deve mostrar na tela a ordem deles após Will retirá-los da mochila.

Entrada

A primeira linha contém um inteiro '**N**' ($1 \leq N \leq 10^3$), representando o número de itens na mochila de Will. A segunda linha consiste de '**N**' inteiros '**I**' ($1 \leq I \leq 10^3$) representando os itens na sequência em que foram colocados na mochila (podem existir itens repetidos).

Saída

A saída consiste de uma linha na qual você deverá imprimir os itens na ordem que Will vai tirá-los da mochila.

Entrada	Saída
5 1 2 3 4 5	5 4 3 2 1
8 9 6 8 1 5 8 3 2	2 3 8 5 1 8 6 9
4 5 7 2 1	1 2 7 5

Ori and the Blind Forest

"Ori And The Blind Forest" conta a história de Ori, um espírito materializado que foi adotado por Naru, uma espécie de urso que há muito tempo vivia em uma floresta encantada. Mas infelizmente ocorre a morte de Naru. Ori se vê sozinho na imensidão verde e assume a responsabilidade de explorar a floresta para sobreviver.



Uma nova versão do jogo está quase completa, faltando apenas implementar uma verificação dos valores de XP obtidos por Ori para saber se ele consegue subir de nível. Sua tarefa será dizer se Ori conseguiu passar de nível, sendo que para isso lhe será dado o XP obtido por ele em cada missão que ele cumpriu, e além disso, será dada uma pontuação bônus para cada missão, onde cada bônus será multiplicado pela pontuação obtida naquela missão.

Entrada

A primeira linha contém um inteiro 'M' ($1 \leq M \leq 1000$), representando o total de XP necessário para subir de nível. A linha seguinte contém um inteiro 'N' ($1 \leq N \leq 1000$), representando a quantidade de missões cumpridas por Ori em determinado mapa de fase. Na linha seguinte serão dados 'N' inteiros 'X' ($1 \leq X \leq 100$), que representam o XP de cada missão. Na última linha serão dados 'N' inteiros 'B' ($1 \leq B \leq 100$), que representam o bônus que deve ser multiplicado pela pontuação da respectiva missão. Ou seja, o primeiro valor de bônus dado, será multiplicado pelo XP da primeira missão, o segundo valor de bônus será multiplicado pelo XP da segunda missão, e assim sucessivamente.

Saída

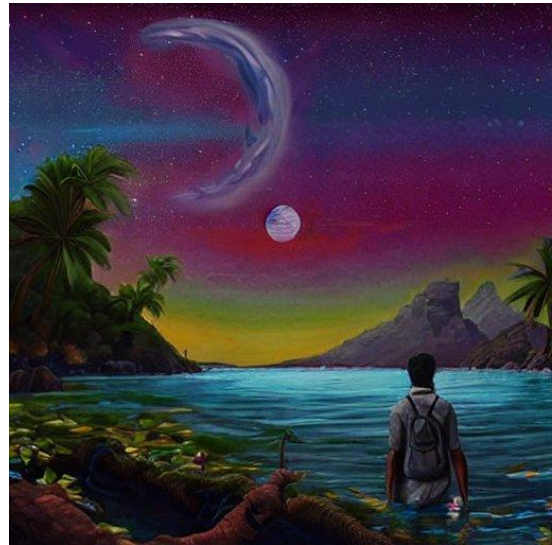
Você deverá imprimir "Upou de Nivel!", se o total de XP for maior ou igual a quantidade 'M' necessária para upar de nível, ou "Nao foi dessa vez!", caso contrário.

Entrada	Saída
15 5 1 2 3 4 5 1 1 1 1 1	Upou de Nivel!
100 4 10 15 10 5 1 2 3 4	Nao foi dessa vez!

Pulando até o tesouro!

Autor: José Roberto Cardoso

Um explorador está em uma emocionante busca por um tesouro escondido em uma ilha misteriosa. Para chegar até o local exato do tesouro, ele precisa atravessar uma série de obstáculos ao longo do caminho. O explorador decidiu que a melhor forma de superar esses obstáculos é criando um mapa. O mapa será representado por um vetor de números inteiros, onde cada elemento representa a altura de um obstáculo no caminho. O explorador precisa escrever um programa que determine a quantidade de obstáculos que ele conseguirá pular, considerando que ele tem uma habilidade máxima de salto.



Escreva um programa que recebe como entrada um vetor de alturas dos obstáculos e a habilidade máxima de salto do explorador, retorne a quantidade máxima de obstáculos que ele conseguirá pular. Ressaltando que obstáculos com altura maior do que a habilidade máxima de salto do explorador não serão ultrapassados e nem mesmo os demais a partir dali.

Entrada

A entrada contém na primeira linha um inteiro “N” ($1 \leq N \leq 100$), representando o número de obstáculos no caminho do explorador. Na próxima linha serão dados “N” inteiros “A” ($1 \leq A \leq 10$), representando as alturas dos “N” obstáculos. Por fim, na última linha teremos um inteiro “S” ($1 \leq S \leq 15$), representando a altura máxima do salto do explorador.

Saída

Imprima na primeira linha a quantidade de obstáculos que o explorador conseguirá pular durante a sua aventura na ilha misteriosa. Na segunda linha imprima “1” se o explorador venceu todos os obstáculos, e “0” (zero), caso contrário.

Entrada	Saída
8 2 3 1 2 4 2 1 3 3	4 0
4 4 2 1 2 4	4 1

Seleção de padawans

Há muito tempo atrás, em uma galáxia muito muito distante, o império do mal estava aumentando sua força, destruindo tudo em seu caminho. Mas os rebeldes não têm medo, ao lado negro da força eles nunca vão se juntar. Uma esperança ainda existe.



Yoda vai nos levar para a luz, é hora de levantar e lutar. Ele voa entre as estrelas na velocidade da luz para ensinar os jovens Jedi. Parabéns, você nível subiu! Agora você pode mestre Jedi ser. Ajudar Mestre Yoda a percorrer a galáxia você precisa. Ajude Mestre Yoda a selecionar novos padawans.

Entrada

A primeira linha contém um inteiro '**N**' ($1 \leq N \leq 10^4$) que é a quantidade de jovens de todas as partes da galáxia interessados em ser um jovem Jedi. A seguir serão dados '**N**' inteiros que representam a força que existe em cada jovem aprendiz. Assim cada um dos '**N**' jovens tem uma força e um identificador, sendo que o primeiro aprendiz tem o identificador **0 (zero)** e o último tem o identificador '**N**' - 1. Na última linha, será dado um inteiro '**F**' ($1 \leq F \leq 10^4$) que representa a força do Mestre Yoda. Ele irá selecionar todos aqueles que tenham pelo menos a metade da força dele.

Saída

Imprima os identificadores dos Padawans que serão selecionados, em ordem crescente. Sempre haverá pelo menos um selecionado.

Entrada	Saída
6 25 20 10 13 19 17 36	0 1 4
4 10 20 30 40 60	2 3
3 20 30 40 70	2
3 20 25 22 45	1

Treino CS:GO



Thiago decidiu treinar em um mapa do jogo de CS:GO. Neste mapa, o objetivo é estourar balões para obter pontos. Cada cor de balão tem uma pontuação aleatória única durante o treino, ou seja, os balões verdes podem ter a pontuação 2, os azuis 4. Porém, uma cor aleatória retira pontos ao invés de acumular. Como Thiago não sabe a cor a princípio, ele deseja saber qual foi sua pontuação ao final do treino.

Entrada

A entrada possui três linhas. A primeira linha contém um inteiro '**N**' ($1 \leq N \leq 1000$), representando o número de balões que Thiago estourou. A segunda linha contém uma sequência de '**N**' inteiros '**P**' ($1 \leq P \leq 100$), representando a pontuação de cada balão estourado. Na última linha será dado um inteiro '**C**' ($1 \leq C \leq 100$), que representa a cor (pontuação) do balão que retira pontos, ou seja, a pontuação a ser retirada toda vez que um balão daquela cor for estourado. Lembre-se que os balões de uma mesma cor tem sempre a mesma pontuação.

Saída

A saída consiste em uma única linha contendo a pontuação final.

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

AVATAR: HABILIDADES DOS DOBRADORES

Autora: Luana Teixeira

Em "Avatar: A Lenda de Aang", os personagens (dobradores) possuem habilidades diferentes para controlar os elementos: água, terra, fogo e ar. Escreva um programa que receba como entrada as habilidades e os identificadores de "N" dobradores. Em seguida, o programa deve exibir os identificadores dos dobradores de um determinado elemento escolhido por Aang.



Entrada

A entrada é composta primeiramente por um inteiro "N" ($1 \leq N \leq 100$), representando o número de dobradores. Na próxima linha serão dados "N" inteiros "E" ($1 \leq E \leq 4$), representando a habilidade de cada dobrador com relação a um dos elementos, sendo 1 representando um dobrador da água, 2 da terra, 3 do fogo e 4 do ar. Na próxima linha serão dados "N" inteiros "X" ($1 \leq X \leq N$), representando os identificadores dos "N" dobradores, que são **identificadores únicos**. Por fim, na última linha será dado um inteiro "Y" ($1 \leq Y \leq 4$), que representa o elemento escolhido por Aang, sendo 1 para água, 2 para terra, 3 para fogo e 4 para ar.

Saída

A saída é composta pelos identificadores dos dobradores do elemento escolhido por Aang, na ordem dada na leitura, todos em uma única linha, separados por espaço. Caso não haja dobradores do elemento escolhido, imprima a frase "**Nenhum**".

Entrada	Saída
6 1 2 2 4 1 3 1 2 3 4 5 6 2	2 3
8 2 3 4 1 3 1 2 1 4 3 2 1 7 6 5 8 1	1 6 8