

TAA - LEE 07

A. Fibonacci

2 seconds, 256 megabytes

Os números de Fibonacci são definidos pela seguinte relação de recorrência:

$$F(n) = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ F(n-1) + F(n-2) & \text{se } n > 1 \end{cases}$$

Assim, sua tarefa é escrever um programa para calcular o n -ésimo número de Fibonacci.

Input

A primeira linha da entrada contém um número T ($1 \leq T \leq 10000$) que indica a quantidade de casos de testes. As próximas T linhas contém um inteiro N ($0 \leq N \leq 10000$) que é o índice do número de Fibonacci a ser calculado.

Output

A saída deve conter T linhas, cada uma contendo o número de Fibonacci correspondente ao índice N , conforme os exemplos.

input
1 5
output
5

input
2 9 4
output
34 3

Como os números podem ser muito grandes, responder com o resto por $10^9 + 7$.

B. Mangos

1 second, 1024 megabytes

Depois da aula, você está na fila do Mango pronto para comprar alguns itens e almoçar. Porém, na hora de pagar, você percebe que seu celular está descarregado e você não está com o cartão para pagar. Recorrendo à sua carteira, você encontra um monte de moedas, mas não tem certeza se elas são suficientes para pagar exatamente o valor da sua compra. Como você sabe que ninguém mais hoje usa dinheiro, você pensa se, dadas as suas moedas, é possível pagar exatamente o valor da compra?

Input

A primeira linha da entrada possui dois inteiro M ($1 \leq M \leq 10^5$) e C ($1 \leq C \leq 1000$), separados por um espaço, indicando o montante da compra e a quantidade de moedas que você possui. A segunda linha contém C inteiros V_i ($1 \leq V_i \leq 10^5$), separados por um espaço, indicando o valor de cada moeda que você possui.

Output

A saída deve conter uma única linha com a palavra "SIM" se for possível pagar exatamente o valor da compra, ou "NAO" caso contrário.

input
74 10 50 25 10 10 5 1 1 1 1 1

output
SIM

input
16 4 10 1 5 25
output
SIM

input
30 4 50 25 10 1
output
NAO

C. Troco

1 second, 256 megabytes

Na hora de pagar o almoço no Mangos, você acabou desistindo de usar as moedas para pagar, porque achou dinheiro perdido no bolso da sua calça.

O funcionário do restaurante, que não estava muito feliz com as suas moedas, resolveu dar o "troco". E de maneira literal, porque ele vai te devolver tudo em moedas.

Pensando na tristeza de ter que lidar com tantas moedas, você pediu a este funcionário que, ao menos se for devolver tudo em moedas, que seja o mínimo possível delas. Então, o funcionário sabendo que você é do curso de tecnologia, pediu para que você calculasse e informasse quantas moedas dariam nesse caso.

Input

A primeira linha da entrada possui dois inteiros M ($1 \leq M \leq 10^5$) e C ($1 \leq C \leq 100$), separados por um espaço, indicando o montante da compra e os tipos de moedas que há no caixa. A segunda linha contém C inteiros V_i ($1 \leq V_i \leq 10^5$), separados por um espaço, indicando o valor de cada tipo diferente de moeda que o restaurante possui.

Output

A saída deve conter uma única linha com um inteiro, indicando o número mínimo de moedas que o funcionário deve te devolver como troco. Se não for possível devolver o troco, deve ser impresso a palavra "impossivel".

input
74 5 50 25 10 5 1
output
7

input
16 4 10 1 5 25
output
3

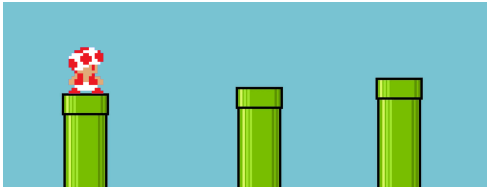
input
10 3 7 5 1
output
2

input
10 3 3 6 8
output
impossivel

D. Toady

1 second, 256 megabytes

Em uma fase do jogo Super Toady World, há N canos numerados de $1, 2, \dots, N$, onde a altura do cano i é h_i .



Toady começa a fase no cano 1, e para terminar a fase, ele pode executar a seguinte ação um número de vezes até alcançar o cano N :

- Se Toady está no cano i , ele pode pular para o cano $i + 1$ ou o cano $i + 2$, ao custo de $|h_i - h_j|$ de stamina, onde j é o cano onde Toady vai parar.

Seu objetivo é encontrar o custo mínimo total necessário de stamina para Toady passar de fase (chegar até o cano N).

Input

A primeira linha contém um inteiro N ($2 \leq N \leq 10^5$), que são a quantidade de canos da fase. A segunda linha contém N inteiros h_i ($1 \leq h_i \leq 10^4$), separados por um espaço cada, que são as alturas dos canos.

Output

A saída deve conter uma única linha com um inteiro, indicando o custo mínimo total necessário para Toady passar de fase.

input
4 10 30 40 20
output
30

input
2 10 10
output
0

input
6 30 10 60 10 60 50
output
40

No primeiro caso de testes, Toady salta os canos $1 \rightarrow 2 \rightarrow 4$ e o custo para chegar ao final é: $|10 - 30| + |30 - 20| = 30$

E. After de Prêmios

2 seconds, 1024 megabytes

O final do semestre chegou e o departamento de Incentivo De Prêmios resolveu premiar os alunos das mais diferentes formas possíveis.

Problems - Codeforces

Durante o after do IDP, um aluno será sorteado e ele terá acesso a uma sala de prêmios e uma sacola gigante. Nessa sala, ele poderá colocar quantos prêmios ele conseguir dentro da sacola, desde que o peso dela suporte, e estes serão dele.

Como cada prêmio tem um peso e um valor específico, a organização do after quer saber qual o máximo prêmio possível um aluno pode conseguir.

Input

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 200$), que indica a quantidade de presentes na sala. A segunda linha contém N inteiros V_i ($0 \leq V_i \leq 10^5$), representando os valores dos presentes. A terceira linha contém N inteiros P_i ($0 \leq P_i \leq 10^6$), onde cada P_i indica o peso do presente correspondente de valor V_i . Por fim, a última linha da entrada contém um inteiro W ($1 \leq W \leq 10^6$), que representa o peso máximo que a sacola suporta.

Output

A saída deve conter um único inteiro, que representa o valor máximo de prêmios que o aluno pode levar, respeitando o limite de peso da sacola.

input
3 30 50 60 3 4 5 8
output
90

input
5 100 100 100 100 100 1 1 1 1 1 5
output
500

input
6 5 6 4 6 5 2 6 5 6 6 3 7 15
output
17

F. Xadrez Bloqueado

1 second, 256 megabytes

No IDP, diversos alunso adora jogar xadrez durante o intervalo das aulas (espero que seja só durante o intervalo). Apesar disso, no final do semestre os alunos estão mais preocupados com provas e trabalhos e acabam não jogando tanto quanto gostariam.

Sem encontrar nenhum parceiro para jogar, e sem mais provas para estudar, Hemersom decidiu criar um jogo utilizando o próprio tabuleiro de xadrez e algumas peças. O jogo funciona da seguinte maneira:

- O tabuleiro é um quadrado com L linhas e C colunas, nemeradas a partir do par de coordenadas $(1, 1)$, no canto superior esquerdo do tabuleiro;
- O jogador coloca peças aleatoriamente no tabuleiro, de forma que essas peças são bloqueios e não podem ser removidas;
- O jogador joga com somente uma peça, que inicia na posição $(1, 1)$;
- O objetivo é chegar na posição (L, C) ;
- As posições iniciais e finais nunca terão bloqueios;
- A peça do jogador pode se mover para baixo ou direita, uma casa por vez, e não pode passar por cima de bloqueios; e
- O jogo termina quando o jogador chega na posição (L, C) ;

Dadas as regras e olhando para o tabuleiro, Hemersom percebeu que existem diversas maneiras de se chegar na posição final. Agora ele está curioso para saber, dada uma configuração do tabuleiro, quantas maneiras existem de se chegar na posição final.

Input

A entrada possui um único caso de teste. A primeira linha contém dois inteiros L e C ($2 \leq L, C \leq 1000$), representando o número de linhas e colunas do tabuleiro, respectivamente. As próximas L linhas contêm C caracteres cada, representando o tabuleiro. Cada caractere pode ser um underline ($_$) ou um hashtag ($\#$), indicando, respectivamente, uma casa livre ou um bloqueio.

Output

A saída deve conter um único inteiro, representando o número de maneiras de se chegar na posição final. Como o número de maneiras pode ser muito grande, o resultado deve ser impresso módulo $10^9 + 7$. E caso não seja possível chegar na posição final, a resposta deve ser -1 .

input
3 4 _ _ # # _ _ _
output
3

input
5 5 _ _ # _ _ _ # _ # _ _ _ # _
output
24

input
10 10 _
output
48620

Cartas: 5 3 2 9 4 1

Entre 5 e 1, escolhe 5 para pontuar
Entre 3 e 1, escolhe 1 para descartar
Entre 3 e 4, escolhe 4 para pontuar
Entre 3 e 9, escolhe 3 para descartar
Entre 2 e 9, escolhe 9 para pontuar
E finalmente escolhe 2 para descartar
Soma: 18

A resposta correta é 18.

Como a conferência manual de que um jogador alcançou o maior número de pontos pode ser um pouco demorado, pediram a sua ajuda para criar um programa que calcule a maior soma de pontos possível para uma sequência específica de cartas.

Input

A entrada do problema consiste em um único caso de teste. A primeira linha contém um inteiro N ($2 \leq N \leq 10^3$), representando o número de cartas disponíveis. A segunda linha contém N inteiros C_i ($1 \leq C_i \leq 10^5$), representando os valores das cartas, na ordem em que elas são distribuídas ao jogador.

Output

A saída deve conter um único inteiro, representando a maior soma de pontos possível que o jogador pode alcançar com a sequência de cartas fornecida.

input
6 5 3 2 9 4 1
output
18

input
5 2 2 2 2 2
output
4

input
5 1 2 3 4 5
output
9

História fictícia.

G. Descartes

1 second, 256 megabytes

Jogos de cartas fazem muito sucesso no IDP e de maneira totalmente ilegal, está acontecendo um cassino clandestino no prédio abandonado do estacionamento, todas os finais de semana, onde alunos, professores e funcionários se reúnem para jogar e apostar nos jogos mais variados possíveis.

Um dos jogos mais populares do cassino é um jogo de cartas chamado Descartes. Esse jogo funciona da seguinte maneira: Dada uma sequência de cartas numeradas e em uma ordem específica, o jogador deve ir escolhendo cartas alternadamente sendo uma carta para pontuar, e em seguida uma carta para ser descartada das pontas das cartas distribuídas. Quando não restarem mais cartas na mesa, a pontuação final será a soma das cartas escolhidas pelo participante.

O jogador vence o jogo se a soma das cartas for a maior possível dada a disposição inicial das cartas ao jogador.

Exemplo de uma rodada com 6 cartas:

[Codeforces](#) (c) Copyright 2010-2025 Mike Mirzayanov
The only programming contests Web 2.0 platform