

Competidor(a): _____

Número de inscrição: _____ – _____ (opcional)

Este Caderno de Tarefas não pode ser levado para casa após a prova. Após a prova entregue este Caderno de Tarefas para seu professor guardar. Os professores poderão devolver os Cadernos de Tarefas aos competidores após o término do período de aplicação das provas (1 a 3 de Junho de 2023).



Olimpíada Brasileira de Informática

OBI2023

Caderno de Tarefas

Modalidade Programação • Nível 1 • Fase 1

1 a 3 de Junho de 2023

A PROVA TEM DURAÇÃO DE 2 horas

Promoção:



Sociedade Brasileira de Computação

Apoio:



Coordenação:



Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

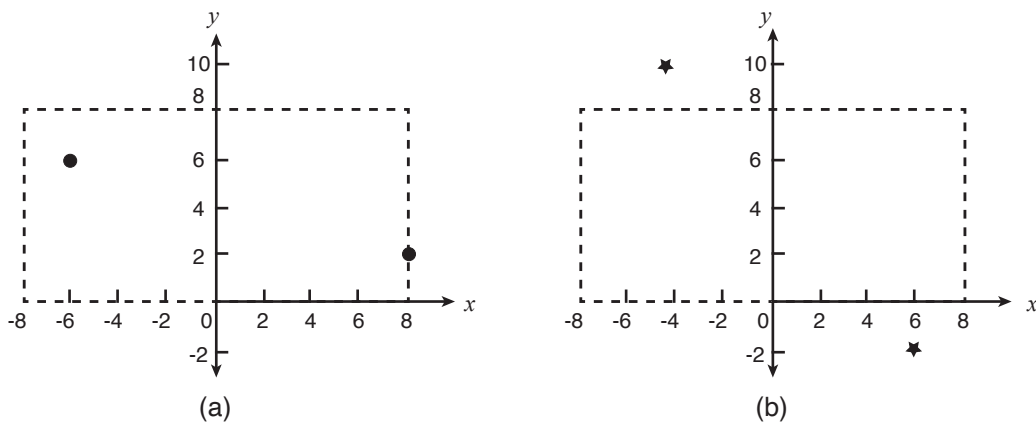
- Este caderno de tarefas é composto por 7 páginas (não contando a folha de rosto), numeradas de 1 a 7. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa; em particular, seu programa não deve escrever frases como “Digite o dado de entrada:” ou similares.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão necessariamente ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 3 devem ser arquivos com sufixo *.py*; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*.
- Na linguagem Java, **não** use o comando *package*, e note que o nome de sua classe principal deve usar somente letras minúsculas (o mesmo nome do arquivo indicado nas tarefas).
- Você pode submeter até 50 soluções para cada tarefa. A pontuação total de cada tarefa é a melhor pontuação entre todas as submissões. Se a tarefa tem sub-tarefas, para cada sub-tarefa é considerada a melhor pontuação entre todas as submissões.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
 - em C: *scanf*, *getchar*, *printf*, *putchar*;
 - em C++: as mesmas de C ou os objetos *cout* e *cin*.
 - em Java: qualquer classe ou função padrão, como por exemplo *Scanner*, *BufferedReader*, *BufferedWriter* e *System.out.println*
 - em Python: *read*, *readline*, *readlines*, *input*, *print*, *write*
 - em Javascript: *scanf*, *printf*
- Procure resolver a tarefa de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

VAR

Nome do arquivo: `var.c`, `var.cpp`, `var.java`, `var.js` ou `var.py`

Com a crescente popularidade dos campeonatos de “Beach Tennis”, uma empresa está desenvolvendo um aplicativo para celular para ser usado por juízes. O objetivo é que, depois de configurar o aplicativo, um “juiz de vídeo” possa usar a câmera do celular para determinar se o impacto da bola com o piso foi dentro ou fora do campo de jogo.

O campo de jogo é um retângulo de dimensões 16m x 8m. A coordenada $(0,0)$ é a posição do juiz, como mostrado na figura (a) abaixo. A figura (a) também mostra duas marcações de bolas dentro do campo de jogo (círculos pretos), nas coordenadas $(-6,6)$ e $(8,2)$. Note que uma bola em cima da linha é considerada dentro do campo de jogo.



A figura (b) mostra duas marcações de bolas fora do campo de jogo (estrelas pretas), nas coordenadas $(-4, 10)$ e $(6, -2)$.

Você foi contratado para testar o novo aplicativo. Como é ainda um protótipo, apenas coordenadas de valores inteiros serão testadas.

Escreva um programa que, dada a coordenada de uma marcação identificada pelo aplicativo, determine se a marcação está dentro ou fora do campo de jogo.

Entrada

A primeira linha da entrada contém um inteiro X , a coordenada x da marcação. A segunda linha contém um inteiro Y , a coordenada y da marcação.

Saída

Seu programa deve produzir uma única linha, contendo um único caractere, que deve ser a letra maiúscula ‘S’ se a marcação está dentro do campo; se a marcação está fora do campo de jogo a linha deve conter a letra maiúscula ‘N’.

Restrições

- $-100 \leq X \leq 100$
- $-100 \leq Y \leq 100$

Informações sobre a pontuação

- A tarefa vale 100 pontos.

Exemplos

Exemplo de entrada 1 6 -2	Exemplo de saída 1 N
--	--------------------------------

Exemplo de entrada 2 8 2	Exemplo de saída 2 S
---------------------------------------	--------------------------------

Estoque

Nome do arquivo: `estoque.c`, `estoque.cpp`, `estoque.java`, `estoque.js` ou `estoque.py`

Você foi contratado(a) para desenvolver um programa de controle de estoque, para uma loja de roupas que está iniciando vendas online. A loja mantém um estoque de roupas, em que cada peça de roupa é identificada por um tipo (por exemplo camisa, calça, saia, vestido, ...) e um tamanho (por exemplo bebê, infantil, pequeno, médio, ...).

O estoque da loja pode ser visto como uma tabela em que cada linha representa um tipo de roupa e cada coluna representa um tamanho, como mostrado na figura (a) abaixo. Na figura, tipos de roupa são representados por números de 1 a 4 e tamanhos são representados por números de 1 a 3.

		TAMANHO		
		1	2	3
T I P O	1	5	2	2
	2	6	4	0
	3	2	1	4
	4	1	3	2

(a)

		TAMANHO		
		1	2	3
T I P O	1	4	2	2
	2	6	4	0
	3	2	1	4
	4	1	3	2

(b)

Assim, a tabela da figura (a) mostra que o estoque da peça de roupa de tipo 1 e tamanho 1 é 5 unidades, e o estoque da peça de roupa de tipo 4 e tamanho 2 é 3 unidades.

Quando uma peça de roupa é vendida, o estoque deve ser atualizado. Por exemplo, se uma peça de roupa de tipo 1 e tamanho 1 for vendida, o estoque atualizado é mostrado na figura (b). Se o estoque para um tipo e tamanho de peça de roupa tem valor zero, peças de roupa desse tipo e tamanho não podem ser vendidas (por exemplo a peça de roupa de tipo 2 e tamanho 3 na figura). Ou seja, a venda não é efetivada.

Dados o estoque inicial e a lista de pedidos de clientes, escreva um programa para determinar quantas peças de roupa são efetivamente vendidas no total. Cada pedido se refere a uma única peça de roupa. As vendas são processadas sequencialmente, na ordem em que os pedidos foram feitos. Se uma venda não é possível por falta de estoque, o pedido correspondente é ignorado.

Entrada

A primeira linha da entrada contém dois números inteiros M e N , indicando respectivamente o número de tipos e o número de tamanhos de peças de roupa no estoque. Tipos são identificados por inteiros de 1 a M e tamanhos são identificados por inteiros de 1 a N . Cada uma das M linhas seguintes contém N inteiros $X_{i,j}$, indicando a quantidade de roupas do tipo i e tamanho j , para $1 \leq i \leq M$ e $1 \leq j \leq N$. A seguir a entrada contém uma linha com um número inteiro P , o número de pedidos recebidos pela loja. Cada uma das P linhas seguintes contém dois inteiros I e J representando respectivamente o tipo e o tamanho da peça de roupa de um pedido. Os pedidos são dados na ordem em que foram feitos.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, o número total de peças de roupas efetivamente vendidas.

Restrições

- $1 \leq M \leq 500$
- $1 \leq N \leq 500$
- $0 \leq X_{i,j} \leq 10$ para $1 \leq i \leq M$ e $1 \leq j \leq N$
- $1 \leq P \leq 1\,000$
- $1 \leq I \leq M$
- $1 \leq J \leq N$

Informações sobre a pontuação

- A tarefa vale 100 pontos.
- Para um conjunto de casos de testes valendo 19 pontos, há apenas um tipo de roupa, ou seja $M = 1$.
- Para um conjunto de casos de testes valendo 17 pontos, há apenas um tamanho de roupa, ou seja $N = 1$.
- para um conjunto de casos de testes valendo os 64 pontos restantes, nenhuma restrição adicional.

Exemplos

<p>Exemplo de entrada 1</p> <pre> 4 3 5 2 2 6 4 0 2 1 4 1 3 2 2 1 1 2 3 </pre>	<p>Exemplo de saída 1</p> <pre> 1 </pre>
<p>Exemplo de entrada 2</p> <pre> 1 4 1 3 2 5 4 1 3 1 3 1 3 1 4 </pre>	<p>Exemplo de saída 2</p> <pre> 3 </pre>

Subsequência

Nome do arquivo: subsequencia.c, subsequencia.cpp, subsequencia.java, subsequencia.js
ou subsequencia.py

Você foi contratado pela Agência Extra-Espacial Brasileira, que procura indícios de vida extra-terrestre.

Um dos telescópios da Agência, para o espectro ultravioleta, gera uma sequência de valores inteiros positivos que devem ser analisados continuamente. Dadas duas sequências S_A e S_B , sua primeira missão é determinar se S_B é uma *subsequência* de S_A .

Uma *subsequência* de uma dada sequência S é um conjunto de elementos de S que não são necessariamente adjacentes mas que mantêm a mesma ordem em que aparecem em S . Por exemplo, [2], [1, 4], [1, 2, 4] e [1, 2, 3, 4] são subsequências de [1, 2, 3, 4], mas [4, 3], [3, 4, 1] e [1, 3, 5] não são.

Entrada

A primeira linha contém dois inteiros A e B , o número de elementos das sequências. A segunda linha contém A inteiros X_i , os números da sequência S_A . A seguir a entrada contém B inteiros Y_i , os números da sequência S_B .

Saída

Seu programa deve produzir uma única linha, contendo um único caractere, que deve ser a letra maiúscula ‘S’ se S_B é uma subsequência da S_A ou a letra maiúscula ‘N’ caso contrário.

Restrições

- $1 \leq A \leq 10^5$
- $1 \leq B \leq A$
- $-10^9 \leq X_i \leq 10^9$ para $1 \leq i \leq A$
- $-10^9 \leq Y_i \leq 10^9$ para $1 \leq i \leq B$

Informações sobre a pontuação

- A tarefa vale 100 pontos.
- Para um conjunto de casos de testes valendo 11 pontos, $A = B = 2$.
- Para um conjunto de casos de testes valendo outros 33 pontos, os números aparecem no máximo uma vez em cada sequência, $A \leq 100$, $1 \leq X_i \leq 100$ e $1 \leq Y_i \leq 100$.
- Para um conjunto de casos de testes valendo outros 56 pontos, nenhuma restrição adicional.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
5 3 1 2 3 4 5 2 3 5	S

Exemplo de entrada 2	Exemplo de saída 2
5 4 8 17 8 21 23 8 8 21 22	N