

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 (13 a 15 de Junho de 2022).



Olimpíada Brasileira de Informática

OBI2022

Caderno de Tarefas

Modalidade Programação • Nível 1 • Fase 1

13 a 15 de Junho de 2022

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 6 páginas (não contando a folha de rosto), numeradas de 1 a 6. 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 Pascal devem ser arquivos com sufixo *.pas*; 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 Pascal: *readln*, *read*, *writeln*, *write*;
 - 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.

Show

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

Um grupo de amigos quer comprar ingressos para um show da sua banda preferida. O show acontece num teatro que tem N filas de assentos, cada fila com M assentos.

Os amigos querem comprar ingressos de forma que os assentos dos amigos:

- sejam todos na mesma fila,
- sejam contíguos (ou seja, um vizinho ao outro) e
- sejam na fila mais próxima possível do palco.

Dado um mapa descrevendo os assentos disponíveis, ajude os amigos a encontrarem os ingressos de acordo com as condições acima.

Entrada

A primeira linha contém três inteiros A , N e M , indicando respectivamente o número de amigos, o número de filas de assentos e o número de assentos em cada fila do teatro. As filas são numeradas de 1 (mais próxima do palco) até N (mais distante do palco). Cada uma das N linhas seguintes contém M inteiros X_i , que podem ter o valor 1 (representando um assento ocupado) ou o valor 0 (representando um assento não ocupado). As filas de assentos são dadas da fila mais distante para a fila mais próxima do palco. Ou seja, a primeira fila dada na entrada é a fila N (mais distante do palco), a última fila dada na entrada é a fila 1 (mais próxima do palco).

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, que deve ser o número da fila em que os amigos conseguem comprar os ingressos se é possível encontrar ingressos de acordo com as condições dadas, ou -1 caso contrário.

Restrições

- $2 \leq A \leq 100$
- $1 \leq N \leq 100$
- $1 \leq M \leq 100$
- $0 \leq X_i \leq 1$ para $1 \leq i \leq M$

Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 20 pontos, $N = 1$.

Exemplo de entrada 1	Exemplo de saída 1
4 3 5 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0	1

Explicação do exemplo 1: São 4 amigos e o teatro tem 3 filas com 5 cadeiras cada fila. A melhor opção é a fila 1, embora os quatro amigos possam também comprar os ingressos na fila 3. Na fila 2 não há cadeiras vagas suficientes.

Exemplo de entrada 2	Exemplo de saída 2
2 3 5 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0	-1

Explicação do exemplo 2: São 2 amigos e o teatro tem 3 filas com 5 cadeiras cada fila. Nenhuma fila tem 2 cadeiras vagas contíguas.

Exemplo de entrada 3	Exemplo de saída 3
3 6 4 0 0 0 0 1 0 0 0 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1	5

Explicação do exemplo 3: São 3 amigos e o teatro tem 6 filas com 4 cadeiras cada fila. As únicas filas com cadeiras vagas suficientes são a fila 5 e a fila 6, então a melhor opção é a fila 5.