

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.

Cinema

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

Duas amigas estão na fila para comprar ingressos para uma sessão de cinema. O preço dos ingressos, em Reais, é dado na tabela abaixo:

Idade	Preço
até 17 anos	15
18 a 59 anos	30
60 anos ou mais	20

Dadas as idades das amigas, escreva um programa para calcular o total a ser pago pelos dois ingressos.

Entrada

A entrada contém duas linhas, cada linha contendo um inteiro, a idade de uma das amigas.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, que deve ser o valor total em Reais a ser pago pelos dois ingressos.

Restrições

- $1 \leq idade \leq 100$

Exemplo de entrada 1	Exemplo de saída 1
100 10	35

Explicação do exemplo 1: Os valores dos ingressos para as idades 100 e 10 são respectivamente 20 e 15, portanto o total é 35.

Exemplo de entrada 2	Exemplo de saída 2
17 18	45

Explicação do exemplo 2: Os valores dos ingressos para as idades 17 e 18 são respectivamente 15 e 30, portanto o total é 45.

Hotel

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

O hotel da Colônia de Férias dos Professores está com uma promoção para as férias de julho. A promoção é válida para quem chegar a partir do dia 1 de julho e sair no dia 1 de agosto.

O preço da diária do hotel é menor para quem chegar mais cedo, e vai aumentando a cada dia. Mais precisamente, a promoção funciona assim:

- A diária do hotel para cada quem chegar no dia 1 é D Reais. Assim, quem chegar no dia 1 vai pagar um total de $31 \times D$ Reais.
- A diária do hotel aumenta A reais por dia. Ou seja, a diária é $D + A$ Reais para quem chegar no dia 2; $D + 2 \times A$ Reais no dia 3; $D + 3 \times A$ Reais no dia 4 e assim por diante.
- A partir do dia 16 a diária não aumenta mais.

Note que quem chegar no dia 2 vai pagar um total de $30 \times (D + A)$ reais; quem chegar no dia 3 vai pagar um total de $29 \times (D + 2 \times A)$ reais, e assim por diante.

Bruno gosta muito da professora Vilma, e para agradá-la quer ajudá-la a planejar suas férias, escrevendo um programa para calcular o total (em Reais) que a professora Vilma vai gastar, dependendo do dia em que chegar no hotel.

Entrada

A primeira linha contém um inteiro D , o valor da diária no dia 1. A segunda linha contém um inteiro A , o aumento da diária a cada dia a partir do dia 2 até o dia 15 (inclusive). A terceira linha contém um inteiro N , o dia de chegada no hotel.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, que deve ser o valor total a ser pago ao hotel pela estadia.

Restrições

- $1 \leq D \leq 1\ 000$
- $1 \leq A \leq 1\ 000$
- $1 \leq N \leq 31$

Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 10 pontos, $N = 1$.

Exemplo de entrada 1	Exemplo de saída 1
100 10 1	3100

Explicação do exemplo 1: Como a chegada é no dia 1, o valor da diária com a promoção é 100. Do dia 1 ao dia 31 são 31 diárias. Assim, o total a pagar é 31×100 .

Exemplo de entrada 2	Exemplo de saída 2
100 20 15	6460

Explicação do exemplo 2: Como a chegada é no dia 15, o valor da diária com a promoção é $100 + 14 \times 20 = 380$. Do dia 15 ao dia 31 são 17 diárias. Assim, o total a pagar é $17 \times 380 = 6460$.

Exemplo de entrada 3	Exemplo de saída 3
100 5 16	2720

Explicação do exemplo 3: Como a chegada é no dia 16, o valor da diária com a promoção é $100 + 14 \times 5 = 170$. Do dia 16 ao dia 31 são 16 diárias. Assim, o total a pagar é $16 \times 170 = 2720$.

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
<pre>4 3 5 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0</pre>	<pre>1</pre>

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.