

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 Sênior • 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 9 páginas (não contando a folha de rosto), numeradas de 1 a 9. 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.

# 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<br>10<br>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  $31 \times 100$ .

| <b>Exemplo de entrada 2</b> | <b>Exemplo de saída 2</b> |
|-----------------------------|---------------------------|
| 100<br>20<br>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  $17 \times 380 = 6460$ .

| <b>Exemplo de entrada 3</b> | <b>Exemplo de saída 3</b> |
|-----------------------------|---------------------------|
| 100<br>5<br>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  $16 \times 170 = 2720$ .

# Maior valor

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

Nesta tarefa, dados três números inteiros  $N$ ,  $M$  e  $S$  você deve escrever um programa para determinar o maior número inteiro  $I$  tal que

- $I$  está dentro do intervalo  $[N, M]$  (ou seja,  $I \geq N$  e  $I \leq M$ ).
- A soma dos dígitos de  $I$  é igual a  $S$ .

## Entrada

A primeira linha contém um inteiro  $N$ , o menor valor do intervalo. A segunda linha contém um inteiro  $M$ , o maior valor do intervalo. A terceira linha contém um inteiro  $S$ , o valor da soma dos dígitos, conforme descrito.

## Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, que deve ser o valor de  $I$  obedecendo às restrições acima, ou  $-1$  se não existir.

## Restrições

- $1 \leq N \leq M \leq 10\,000$
- $1 \leq S \leq 36$

## Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 10 pontos,  $M \leq 100$ .

| Exemplo de entrada 1 | Exemplo de saída 1 |
|----------------------|--------------------|
| 1<br>100<br>6        | 60                 |

*Explicação do exemplo 1:* 60 é o maior inteiro no intervalo  $[1, 100]$  cuja soma dos dígitos é igual a 6.

| Exemplo de entrada 2 | Exemplo de saída 2 |
|----------------------|--------------------|
| 1000<br>1001<br>3    | -1                 |

*Explicação do exemplo 2:* Não há número inteiro no intervalo  $[1000, 1001]$  cuja soma dos dígitos é igual a 3.

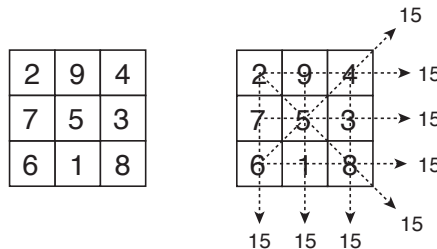
| Exemplo de entrada 3 | Exemplo de saída 3 |
|----------------------|--------------------|
| 80<br>500<br>12      | 480                |

*Explicação do exemplo 3:* 480 é o maior inteiro no intervalo  $[80, 500]$  cuja soma dos dígitos é igual a 12.

# Quadrado Mágico

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

Em um Quadrado Mágico, a soma de qualquer coluna, linha ou diagonal tem sempre o mesmo valor, e nenhum número aparece mais do que uma vez.



A *dimensão* de um quadrado mágico é o número de colunas (ou de linhas, já que o número de colunas é igual ao número de linhas).

Rita encontrou um caderno antigo de sua avó, repleto de quadrados mágicos de todas as dimensões. Infelizmente alguns dos números estão ilegíveis. Você pode ajudá-la?

Dado um quadrado mágico com exatamente um número ilegível, determine o valor e a posição desse número.

## Entrada

A primeira linha da entrada contém um número inteiro  $N$ , a dimensão do quadrado mágico. Cada uma das  $N$  linhas seguintes contém  $N$  inteiros  $X_i$ . Exatamente um dos números do quadrado da entrada é igual a zero, indicando o número ilegível.

## Saída

Seu programa deve produzir três linhas, cada uma contendo um único número inteiro. A primeira linha deve conter o valor do número ilegível. A segunda linha deve conter a linha do número ilegível no quadrado (as linhas do quadrado variam de 1 a  $N$ ). A terceira linha deve conter a coluna do número ilegível no quadrado (as colunas do quadrado variam de 1 a  $N$ ).

## Restrições

- $3 \leq N \leq 10$
- $0 \leq X_i \leq 100$ , para  $1 \leq i \leq N$

## Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 10 pontos,  $1 \leq N \leq 3$ .

| Exemplo de entrada 1 | Exemplo de saída 1 |
|----------------------|--------------------|
| 3                    | 5                  |
| 2 9 4                | 2                  |
| 7 0 3                | 2                  |
| 6 1 8                |                    |

*Explicação do exemplo 1:* O valor do número ilegível é 5 e sua posição é linha 2 e coluna 2.

| <b>Exemplo de entrada 2</b> | <b>Exemplo de saída 2</b> |
|-----------------------------|---------------------------|
| 4                           | 10                        |
| 11 8 5 0                    | 1                         |
| 14 1 4 15                   | 4                         |
| 2 13 16 3                   |                           |
| 7 12 9 6                    |                           |

*Explicação do exemplo 2:* O valor do número ilegível é 10 e sua posição é linha 1 e coluna 4.



# Chuva

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

Eventos climáticos extremos como chuvas descomunais estão cada vez mais frequentes e intensos em todo o mundo.

O Centro Nacional de Monitoramento da Nlogônia tem medidores de quantidade de chuva dia-a-dia espalhados por todo o reino. Cada medição é um número inteiro, indicando a quantidade de chuva, em milímetros, que caiu na Nlogônia num determinado dia. Como o sistema existe há vários anos, a lista de medições é muito grande.

Preocupado com o assunto, o rei da Nlogônia mandou que o Ministro da Ciência crie um programa de computador para calcular quantos intervalos de dias existem na lista de medições tal que a soma das medições nesse intervalo é igual a um certo valor.

Mais precisamente, considere uma lista com  $N$  medições, indicando a quantidade de chuva do dia 1 ao dia  $N$ . Considere ainda todos os possíveis intervalos de dias entre 1 e  $N$ , cada intervalo definido pelo dia inicial e dia final do intervalo. O rei deseja saber quantos intervalos têm a soma das medições exatamente igual a um certo valor  $S$ .

O Ministro da Ciência é um físico brilhante, mas não sabe resolver essa tarefa. Você poderia ajudá-lo?

## Entrada

A primeira linha contém um inteiro  $N$ , o número de medições na lista. A segunda linha contém um inteiro  $S$ , o valor da soma desejada. A terceira linha contém  $N$  inteiros  $X_i$ , os valores das medições.

## Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, que deve ser o número de intervalos que têm a soma das medições igual a  $S$ .

## Restrições

- $1 \leq N \leq 100\,000$
- $0 \leq S \leq 1\,000\,000$
- $0 \leq X_i \leq 10$ , para  $1 \leq i \leq N$

## Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 20 pontos,  $N \leq 300$ .
- Para um outro conjunto de casos de testes valendo 30 pontos,  $N \leq 1000$ .

| Exemplo de entrada 1  | Exemplo de saída 1 |
|-----------------------|--------------------|
| 6<br>2<br>0 2 0 1 0 1 | 6                  |

*Explicação do exemplo 1:* São 6 os intervalos com soma igual a 2: [2], [0,2], [2,0], [0,2,0], [1,0,1] e [0,1,0,1]

| <b>Exemplo de entrada 2</b>  | <b>Exemplo de saída 2</b> |
|------------------------------|---------------------------|
| 8<br>13<br>10 1 0 0 9 10 1 5 | 0                         |

*Explicação do exemplo 2:* Não há intervalo com soma igual a 13.

| <b>Exemplo de entrada 3</b> | <b>Exemplo de saída 3</b> |
|-----------------------------|---------------------------|
| 5<br>6<br>1 0 3 0 2         | 1                         |

*Explicação do exemplo 3:* Há apenas um intervalo com soma igual a 6: [1, 0, 3, 0, 2].