

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 2 • 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 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 $16 \times 170 = 2720$.

Bombom

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

Bombom é um jogo de cartas para duas pessoas, jogado com apenas dezesseis cartas: Ás, Valete, Dama e Rei, nos quatro naipes (Copas, Espadas, Ouros e Paus). Cada carta tem um valor, que depende da figura e do naipe.

A cada partida, as cartas são embaralhadas e colocadas em um monte. Inicialmente uma carta do monte é virada e mostrada aos dois jogadores: o naipe dessa carta é chamado de *naipe dominante* da partida.

Então cada jogador recebe três cartas do monte. Ganha a partida o jogador que tem as cartas cuja soma dos valores é maior.

O valor das cartas é dado na tabela abaixo:

Figura	Naipe não dominante	Naipe Dominante
Ás	10	14
Valete	11	15
Dama	12	16
Rei	13	17

Luana e Edu estão jogando Bombom e querem sua ajuda para determinar o vencedor da partida, ou se há empate.

Entrada

A entrada contém sete linhas, cada linha contendo a descrição de uma carta. Cada carta é descrita por duas letras. A primeira letra de uma carta indica a figura e pode ser A, J, Q ou K, representando respectivamente as figuras Ás, Valete, Dama e Rei. A segunda letra de uma carta indica o naipe e pode ser C, E, O ou P, representando respectivamente os naipes Copas, Espadas, Ouros e Paus. O naipe da primeira carta da entrada é o naipe dominante da partida. A segunda, terceira e quarta cartas da entrada são as cartas de Luana. A quinta, sexta e sétima cartas da entrada são as cartas de Edu.

Saída

Seu programa deve produzir uma única linha, contendo somente o nome do jogador que ganha a partida, ou **empate** caso não haja um ganhador.

Restrições

- As cartas na entrada obedecem ao formato descrito no enunciado.
- Não há cartas repetidas na entrada.

Exemplo de entrada 1	Exemplo de saída 1
AC JC JE JP KO KP QE	Edu

Explicação do exemplo 1: O naipe dominante é Copas. As cartas de Luana valem $15 + 11 + 11 = 37$; as cartas de Edu valem $13 + 13 + 12 = 38$. Assim, Edu é o vencedor.

Exemplo de entrada 2	Exemplo de saída 2
QP QC AC KP KO KE KC	empate

Explicação do exemplo 2: O naipe dominante é Paus. As cartas de Luana valem $12 + 10 + 17 = 39$; as cartas de Edu valem $13 + 13 + 13 = 39$. Assim, há empate.

Exemplo de entrada 3	Exemplo de saída 3
KE QE AC AE AP KO JE	Luana

Explicação do exemplo 3: O naipe dominante é Espadas. As cartas de Luana valem $16 + 10 + 14 = 40$; as cartas de Edu valem $10 + 13 + 15 = 38$. Assim, Luana é a vencedora.

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
1 100 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 1001 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
80 500 12	480

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

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 2 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]

Exemplo de entrada 2	Exemplo de saída 2
8 13 10 1 0 0 9 10 1 5	0

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

Exemplo de entrada 3	Exemplo de saída 3
5 6 1 0 3 0 2	1

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