

Equipe: _____

Programação Competitiva - Desafios

Caderno de Tarefas

04 de dezembro de 2024

A PROVA TEM DURAÇÃO DE 3 HORAS

Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 15 páginas (não contando a capa e folha de rosto), numeradas de 1 a 15. Verifique se o caderno está completo.
- A prova deve ser feita em equipe.
- 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 .py3; 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).
- Para tarefas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada tarefa.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou pen-drive, conforme especificado pelo seu professor.
- 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. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

Cartas

Um novo game de realidade aumentada tem, dentro dele, um mini-game que aparece em certas situações para aumentar o ganho de pontos do game principal. O mini-game é um joguinho de memória com quatro cartas, formando dois pares de cartas iguais. Quer dizer, duas cartas têm um número inteiro N marcado em uma de suas faces e as outras duas cartas têm um outro número inteiro M , $N \neq M$. Neste problema, o jogador já virou três cartas, como mostrado na figura.



Claro que, dadas as condições, a carta que falta virar vai formar par com uma das três que já foram viradas. No caso da figura, o número marcado na carta que ainda falta virar é 11. Implemente um programa que, dados os números de três cartas, imprima o número da carta que ainda falta virar!

Entrada

A primeira linha da entrada contém um inteiro A , representando a primeira carta aberta. A segunda linha contém o inteiro B , representando a segunda carta. A terceira linha contém o inteiro C , que é a terceira carta.

Saída

Seu programa deve imprimir uma linha contendo um inteiro representando o número que está na carta que ainda falta virar.

Restrições

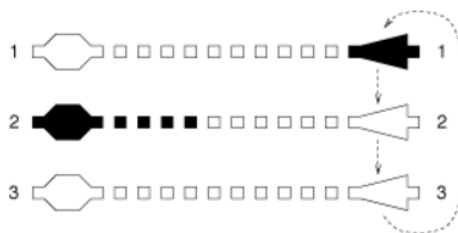
- $0 \leq A, B, C \leq 100$

Exemplos

Entrada 40 11 40	Saída 11
Entrada 8 8 96	Saída 96

Game-10

No princípio dos anos 1980 surgiram nos colégios os primeiros relógios de pulso digitais com joguinhos. Era uma febre entre os alunos e quem tinha um era muito popular na hora do recreio. Os joguinhos eram bem simples, mas muito legais. Um dos primeiros era o Game-10, no qual você controlava um avião que aparecia na parte direita do visor. Na parte esquerda aparecia um disco voador em qualquer uma de três posições, aleatoriamente, e lançava um míssil. O objetivo do jogador era movimentar o avião verticalmente para que ficasse na frente do disco voador (na mesma linha horizontal, do lado direito) e atirar para interceptar o míssil antes que esse atingisse o avião.



Como o movimento do avião era feito com apenas um botão, só dava para movimentar em um sentido: ao apertar o botão sucessivas vezes, o avião se movia na sequência de posições ... $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \dots$

Veja que, na situação da figura, o jogador deveria apertar o botão apenas uma vez, para ir da posição 1 para a posição 2, e conseguir atirar e interceptar o míssil.

Neste problema vamos considerar que existem N posições e não apenas três. Dado o número de posições N , a posição D na qual o disco voador aparece, e a posição A onde está o avião, seu programa deve computar o número mínimo de vezes que o jogador precisa apertar o botão para movimentar o avião até a mesma posição do disco voador e poder atirar!

Entrada

A primeira linha da entrada contém um inteiro N , o número de posições. A segunda linha contém um inteiro D , a posição do disco voador. A terceira linha contém um inteiro A , a posição do avião.

Saída

Seu programa deve imprimir uma linha contendo um inteiro, o número mínimo de vezes que o jogador deve apertar o botão para poder atirar.

Restrições

- $3 \leq N \leq 100$
- $1 \leq D, A \leq N$

Exemplos

Entrada 3 2 1	Saída 1
Entrada 20 8 13	Saída 15

Divisão do Tesouro

O Capitão Olho Roxo e seus marinheiros encontraram uma arca com uma grande quantidade de moedas de ouro idênticas. Para a divisão das moedas, todos concordaram com a seguinte sugestão do Capitão:

- cada marinheiro exceto o Capitão deveria receber exatamente o mesmo número de moedas; e
- o Capitão deveria receber o dobro de moedas que um marinheiro recebe.

Pode ser que o fato de o Capitão ser o único com uma pistola a bordo tenha contribuído para a concordância de todos, mas também contribuiu o fato de que na forma proposta a divisão era perfeita, não sobrando ou faltando moedas.

Dados o número de moedas na arca e o número de marinheiros, escreva um programa para determinar quantas moedas o Capitão Olho Roxo recebeu.

Entrada

A primeira linha da entrada contém um número inteiro A , o número de moedas na arca. A segunda linha contém um inteiro N , o número de marinheiros (não contando o Capitão).

Saída

Seu programa deve produzir na saída uma única linha, contendo um único inteiro, o número de moedas que o Capitão Olho Roxo deve receber.

Restrições

- $3 \leq A \leq 10000$
- $1 \leq N \leq 1000$

Exemplos

Entrada 221 11	Saída 34
Entrada 1000 8	Saída 200
Entrada 3 1	Saída 2

Plano de Internet

João conseguiu contratar um ótimo plano de Internet para o seu telefone celular. O plano permite que João utilize uma quota de até X megabytes de dados por mês para navegar na Internet. Se João não usa toda a sua quota no mês, os megabytes que ele não usou são adicionados à quota do mês seguinte. Pelo contrato, João nunca pode usar mais megabytes do que a sua quota corrente. Por exemplo, se $X=200$ megabytes e João usou 150 no primeiro mês e 220 megabytes no segundo mês, então no terceiro mês João tem uma quota de 230 megabytes para usar (50 megabytes são transferidos do primeiro para o segundo mês, 30 megabytes são transferidos do segundo para o terceiro mês). Nesta tarefa são dados o valor da quota mensal X e quantos megabytes João usou em cada um dos primeiros N meses do plano. Você deve determinar quantos megabytes João tem para usar no mês $N+1$.

Entrada

A primeira linha da entrada contém um número inteiro X , o valor da quota mensal em megabytes. A segunda linha contém um inteiro N , o número de meses. Cada uma das linhas seguintes contém um número inteiro M_i , indicando a quantidade de megabytes que João usou em cada mês, do mês 1 até o mês N .

Saída

Seu programa deve produzir uma única linha, contendo um único número inteiro, a quantidade de megabytes que João tem para usar no mês $N+1$.

Restrições

- $1 \leq X \leq 100$
- $1 \leq N \leq 100$
- $0 \leq M_i \leq 10000$ para $1 \leq i \leq N$
- M_i nunca é maior do que a quantidade de megabytes que João tem para usar no mês.

Informações sobre a pontuação

- Para um conjunto de casos de testes valendo 10 pontos, $1 \leq N \leq 3$.
- Para um conjunto de casos de testes valendo 90 pontos, nenhuma restrição adicional.

Exemplos

Entrada	Saída
100	130
2	
50	

120	
Entrada 10 3 4 6 2	Saída 28
Entrada 100 2 100 100	Saída 100

Chuva

Está chovendo tanto na Obilândia que começaram a aparecer goteiras dentro da casa do prefeito. Uma dessas goteiras está fazendo escorrer água verticalmente, a partir de um ponto no teto, numa parede onde há várias prateleiras horizontais. Quando a água bate em uma prateleira, ela começa a escorrer horizontalmente para os dois lados, direita e esquerda, até as extremidades da prateleira, quando volta a escorrer verticalmente.

Vamos representar a parede por uma matriz de N linhas e M colunas de caracteres, como mostrado abaixo. As prateleiras serão representadas por "#" e a parede por ".". Só existem prateleiras nas linhas pares e elas nunca encostam na borda da parede. Há apenas um ponto de vazamento representado pelo caractere "o" na primeira linha.

```

.....o.....
.###...####.#.
.....
..#####.....
.....
.#.####....##.
.....
.....####.....
.....

```

Para deixar mais rigorosa a forma como a água vai escorrer, seja $c(i,j)$ o caractere na linha i coluna j . Se $c(i,j) = "."$, então ele deve virar "o" sempre que:

- $c(i-1,j) = "o"$; ou
- $c(i,j-1) = "o"$ e $c(i+1,j-1) = "#"$; ou
- $c(i,j+1) = "o"$ e $c(i+1,j+1) = "#"$.

```

.....000000..
.###...o####o#.
.00000000..o..
.o#####o..o..
000.....o.0000
o#o####.o.o##o
o.o.0000000..o
o.o.o####oo..o
o.o.o....oo..o

```

Neste problema, dada a matriz representando a parede no início do vazamento, seu programa deve imprimir na saída uma matriz representando a parede usando o caractere "o" exatamente nas posições que serão molhadas pelo vazamento, como ilustrado acima.

Entrada

A primeira linha da entrada contém dois inteiros N e M, respectivamente o número de linhas e colunas da matriz. As N linhas seguintes da entrada contém, cada uma, uma sequência de M caracteres entre três possíveis: ".", "#" ou "o".

Saída

Seu programa deve imprimir N linhas, cada uma contendo uma sequência de M caracteres, representando a matriz da entrada usando o caractere "o" exatamente nas posições que serão molhadas pelo vazamento.

Restrições

- $3 \leq N \leq 500$ e $3 \leq M \leq 500$;
- O número de linhas N é ímpar;
- Há exatamente um caractere "o" na primeira linha;
- As linhas ímpares, a primeira coluna e a última coluna não possuem o caractere "#".

Exemplos

Entrada 9 14O..... .###...####.#.#####.....#.####...##.####.....	Saída000000.. .###..o####o#. .00000000..O.. .o#####O..O.. 000.....O.0000 O#o####.O.O##O O.O.0000000..O O.O.O####00..O O.O.O....00..O
Entrada 7 5 .O... .#...#..	Saída 000.. O#O.. O.O.. O.O.. 0000.. 00#O.. 00.O..
Entrada 3 3 ..O .#. ...	Saída ..O .#O ..O

Calçada Imperial

Na calçada em frente ao Palácio Imperial, não se sabe a razão, existe uma sequência de N números desenhados no chão. A sequência é composta apenas pelos números de 1 a N . Veja um exemplo na coluna (a) da figura abaixo, para $N=12$.

3	3
2	2
5	5
2	2
10	10
4	4
4	4
7	7
12	12
2	2
8	8
10	10
(a)	(b)

Ninguém sabe o significado da sequência e, justamente por isso, várias teorias malucas surgiram. Uma delas diz que a sequência representa, na verdade, apenas um valor que estaria relacionado a um grande segredo dos imperadores. Esse valor é a quantidade *máxima* de números da sequência que poderiam ser marcados com um círculo, de modo que a sequência de números marcados não contenha dois números iguais consecutivos e seja composta de no máximo dois números distintos.

A coluna (b) da figura ilustra uma sequência de 4 números marcados que obedece a restrição acima. Você consegue verificar que essa é, de fato, a quantidade máxima possível de números numa sequência marcada?

Neste problema, dada a sequência original de números desenhados no chão da calçada, seu programa deve computar e imprimir a quantidade máxima de números da sequência que poderiam ser marcados com um círculo sem que haja dois números iguais consecutivos na sequência marcada e tal que ela seja composta de no máximo dois números distintos.

Entrada

A primeira linha da entrada contém um inteiro N representando o tamanho da sequência. As N linhas seguintes contêm, cada uma, um inteiro V_i , para $1 \leq i \leq N$, definindo a sequência de números desenhados no chão da calçada imperial.

Saída

Seu programa deve imprimir uma linha contendo um número inteiro representando a quantidade máxima de números da sequência que poderiam ser marcados com um

círculo sem que haja dois números iguais consecutivos na sequência marcada e tal que ela seja composta de no máximo dois números distintos.

Restrições

- $1 \leq N \leq 500$
- $1 \leq V_i \leq N$, para $1 \leq i \leq N$

Exemplos

Entrada 1 1	Saída 1
Entrada 12 3 2 5 2 10 4 4 7 12 2 8 10	Saída 4

Código**7**

Gabriel inventou um código para representar números naturais, usando uma sequência de zeros e uns. Funciona assim, o número natural é representado pela quantidade de vezes que o padrão "100" aparece na sequência. Por exemplo, na sequência 11101001010011110, o padrão aparece duas vezes; e na sequência 11101010111110111010101 ele não aparece nenhuma vez. Você deve ajudar Gabriel e implementar um programa que, dada a sequência de zeros e uns, calcule quantas vezes o padrão "100" aparece nela.

Entrada

A primeira linha da entrada contém um inteiro N, o tamanho da sequência. A segunda linha contém a sequência de N zeros e uns, separados por espaço em branco.

Saída

Seu programa deve imprimir um inteiro, quantas vezes o padrão "100" aparece na sequência.

Restrições

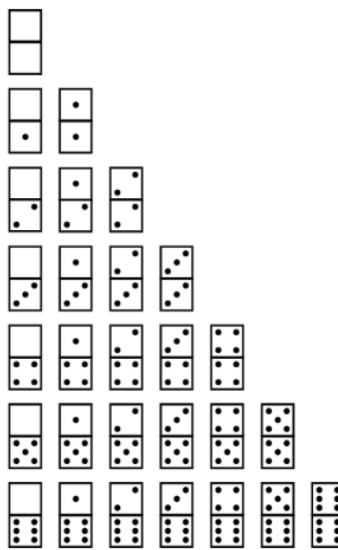
- $3 \leq N \leq 10^4$

Exemplos

Entrada 17 1 1 1 0 1 0 0 1 0 1 0 0 1 1 1 1 0	Saída 2
Entrada 8 1 1 1 1 0 1 1 1	Saída 0
Entrada 3 1 0 0	Saída 1
Entrada 3 0 1 0	Saída 0

Dominó

O jogo de dominó tradicional, conhecido como duplo-6, possui 28 peças. Cada peça está dividida em dois quadrados e dentro de cada quadrado há entre 0 e 6 círculos. O jogo é chamado de duplo-6 justamente porque esse é o maior número de círculos que aparece num quadrado de uma peça. A figura ao lado mostra uma forma de organizar as 28 peças do jogo duplo-6 em 7 linhas. Essa figura permite ver claramente quantas peças haveria num jogo de dominó, por exemplo, do tipo duplo-4: seriam todas as peças das 5 primeiras linhas, 15 peças no total. Também poderíamos ver, seguindo o padrão da figura, quantas peças possui o jogo de dominó conhecido como mexicano, que é o duplo-12. Seriam 91 peças, correspondendo a 13 linhas.



Para a nossa sorte, existe uma fórmula com a qual podemos calcular facilmente o número de peças de um jogo do tipo duplo-N, para um número N natural qualquer: $((N+1)*(N+2))/2$. Neste problema, estamos precisando da sua ajuda para escrever um programa que, dado o valor N, use esta fórmula para calcular e imprimir quantas peças existem num jogo de dominó do tipo duplo-N.

Entrada

A primeira linha da entrada contém um número natural N representando o tipo do jogo de dominó: duplo-N.

Saída

Seu programa deve imprimir uma linha contendo um número natural representando quantas peças existem num jogo de dominó do tipo duplo-N.

Restrições

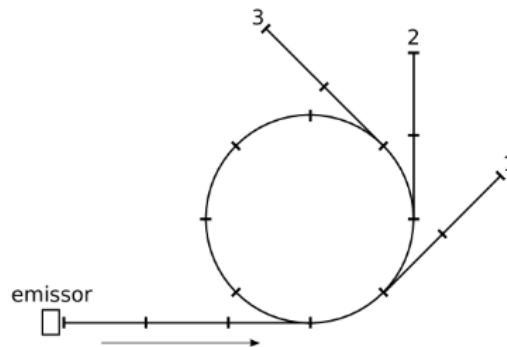
- $0 \leq N \leq 10000$

Exemplos

Entrada 6	Saída 28
Entrada 12	Saída 91

Acelerador de Partículas

A universidade está inaugurando um grande acelerador de partículas, com um emissor e três sensores, numerados 1, 2 e 3. Uma partícula, após sair do emissor, entra no acelerador onde pode dar várias voltas sendo acelerada a velocidades muito altas. Num determinado momento, a partícula sai do acelerador por uma das três saídas, atingindo um dos sensores. A figura mostra o caminho por onde as partículas trafegam, com uma graduação de 1 quilômetro. Por exemplo, do emissor até o acelerador são 3 quilômetros e a circunferência do acelerador tem 8 quilômetros.



Neste problema, será dada a distância total, em quilômetros, percorrida por uma certa partícula trafegando do emissor até algum sensor e seu programa deve determinar qual sensor foi atingido pela partícula. Por exemplo, veja que se a distância total for 23 quilômetros, então a partícula tem que ter atingido o sensor 2.

Entrada

A entrada consiste de apenas uma linha contendo um inteiro D , representando a distância total percorrida pela partícula.

Saída

Restrições

- $6 \leq D \leq 800008$. D sempre será a distância total percorrida entre o emissor e algum sensor.

Exemplos

Entrada 23	Saída 2
Entrada 6	Saída 1
Entrada 9192	Saída 3

Letras

10

Considere as definições abaixo:

- Uma palavra é uma sequência de letras consecutivas.
- Um texto é um conjunto de palavras separadas pelo caractere espaço em branco.

Você foi contratado pela empresa Booble para escrever um programa que, dados uma letra e um texto, determina a porcentagem de palavras do texto que contém a letra dada.

Entrada

A primeira linha da entrada contém um único caractere, a letra de interesse na pesquisa. A segunda linha contém um texto, como definido acima.

Saída

Seu programa deve produzir uma única linha, contendo um único número real, a porcentagem de palavras do texto que contém a letra dada, com precisão de uma casa decimal.

Restrições

- O texto é composto apenas por letras minúsculas e o caractere espaço em branco.
- O texto é formado por no mínimo um caractere, e no máximo 1000 caracteres.
- O texto não contém dois espaços em branco consecutivos

Exemplos

Entrada p papagaio	Saída 100.0
Entrada o no meio do caminho tinha uma pedra tinha uma pedra no meio do caminho	Saída 57.1
Entrada b nunca me esquecerei que no meio do caminho tinha uma pedra	Saída 0.0