

Equipe: _____

Programação Competitiva - Desafios

Caderno de Tarefas

29 de novembro 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 14 páginas (não contando a capa e folha de rosto), numeradas de 1 a 14. 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.

Jogo de Par ou Ímpar

Dois amigos, Alice e Bob, estão jogando um jogo muito simples, em que um deles grita ou "par" ou "ímpar" e o outro imediatamente responde ao contrário, respectivamente "ímpar" ou "par". Em seguida, ambos exibem ao mesmo tempo uma mão cada um, em que alguns dedos estão estendidos e outros dobrados. Então eles contam o número total de dedos estendidos. Se a soma for par, quem gritou "par" ganha. Se a soma for ímpar, quem gritou "ímpar" ganha. Por exemplo, suponhamos que a Alice gritou "par" e o Bob respondeu "ímpar". Em seguida, Alice não deixou nenhum dos seus dedos estendidos, ao passo que Bob deixou três dedos estendidos. A soma então é três, que é ímpar, portanto Bob ganhou. Seu programa deve determinar quem ganhou, tendo a informação de quem gritou par e o número de dedos estendidos de cada um.

Entrada

A entrada contém três linhas, cada uma com um número inteiro, P, D_1 e D_2, nesta ordem. Se P = 0 então Alice gritou "par", ao passo que se P=1 então Bob gritou "par". Os números D_1 e D_2 indicam, respectivamente, o número de dedos estendidos da Alice e do Bob.

Saída

Seu programa deverá imprimir uma única linha, contendo um único número inteiro, que deve ser 0 se Alice foi a ganhadora, ou 1 se Bob foi o ganhador.

Restrições

- P = 0 ou P = 1
- $0 \leq D_1 \leq 5$
- $0 \leq D_2 \leq 5$

Exemplos

Entrada 0 0 3	Saída 1
Entrada 1 0 3	Saída 0
Entrada 0 1 5	Saída 0

Irmãos

Otávio tem dois irmãos, um mais velho (Orlando) e um mais novo do que ele (Oscar). As idades dos três irmãos formam uma *progressão aritmética*: a diferença de idade dos dois irmãos mais novos (Otávio e Oscar) é igual à diferença de idade dos dois irmãos mais velhos (Orlando e Otávio).

Dadas as idades de Otávio e de seu irmão mais novo, escreva um programa para determinar a idade do irmão mais velho.

Entrada

A primeira linha da entrada contém um inteiro N , a idade do irmão mais novo de Otávio. A segunda linha contém um inteiro M , a idade de Otávio.

Saída

Seu programa deve produzir na saída uma única linha, contendo um único número inteiro, a idade do irmão mais velho de Otávio.

Restrições

- $1 \leq N \leq 40$
- $N \leq M \leq 40$

Exemplos

Entrada 13 16	Saída 19
Entrada 14 14	Saída 14

Divisores

Um pesquisador precisa saber o número de divisores de um número dado. Por exemplo, 660 tem 24 divisores: 1, 2, 3, 4, 5, 6, 10, 11, 12, 15, 20, 22, 30, 33, 44, 55, 60, 66, 110, 132, 165, 220, 330, 660.

A fatoração de 660 em fatores primos é: $2^2 \times 3 \times 5 \times 11$. Então o número de divisores é calculado pelo produtos dos expoentes acrescentados de um: $(2 + 1) \times (1 + 1) \times (1 + 1) \times (1 + 1) = 24$.

Um outro exemplo é o número 50, que tem 6 divisores. De fato, $50 = 2 \times 5^2$, portanto o número de divisores é $(1+1) \times (2+1) = 6$.

Entrada

A entrada consiste de uma linha contendo um inteiro N.

Saída

Seu programa deve escrever uma única linha na saída, contendo um único número inteiro, a quantidade de divisores de N.

Restrições

- $1 \leq N \leq 10^4$

Exemplos

Entrada 660	Saída 24
Entrada 50	Saída 6
Entrada 9216	Saída 33

Cartas

Beatriz gosta muito de jogar cartas com as amigas. Para treinar memória e raciocínio lógico, ela inventou um pequeno passatempo com cartas. Ela retira as cinco primeiras cartas do topo de um baralho bem embaralhado, e as coloca em sequência, da esquerda para a direita, na mesa, com as faces voltadas para baixo.

Então ela olha, por um breve instante, cada uma das cartas da sequência (e logo as recoloca na mesa, com a face para baixo). Usando apenas a sua memória, Beatriz deve agora dizer se a sequência de cartas está ordenada crescentemente, decrescentemente, ou não está ordenada.

De tanto jogar, ela está ficando cansada, e não confia em seu próprio julgamento para saber se acertou ou errou. Por isso, ela pediu para você fazer um programa que, dada uma sequência de cinco cartas, determine se a sequência dada está ordenada crescentemente, decrescentemente, ou não está ordenada.

Entrada

A entrada consiste de uma única linha que contém as cinco cartas da sequência. Os valores das cartas são representados por inteiros entre 1 e 13. As cinco cartas têm valores distintos.

Saída

Seu programa deve produzir uma única linha, contendo um único caractere maiúsculo: "C" caso a sequência dada esteja ordenada crescentemente, "D" se estiver ordenada decrescentemente, ou "N" caso contrário.

Restrições

- O valor de cada carta é um inteiro entre 1 e 13.

Exemplos

Entrada 1 2 3 4 5 6	Saída C
Entrada 5 7 10 9 11	Saída N
Entrada 12 10 4 3 2	Saída D

PacMan

Pacman é um jogo muito conhecido, onde o personagem tenta comer a maior quantidade possível de bolinhas, tendo ao mesmo tempo que fugir de vários fantasmas. Dessa vez, nosso personagem quer carregar a comida coletada para casa, mas o encontro com um fantasma, ao invés de terminar o jogo, faz com que toda a comida coletada seja roubada.

Neste problema os fantasmas não se movem, e o jogador sempre faz o Pacman percorrer o seguinte caminho:

1. O Pacman começa no canto superior esquerdo do tabuleiro.
2. O Pacman percorre toda a linha, da esquerda para direita, até chegar ao lado direito do tabuleiro.
3. O jogador desce uma posição, e percorre toda a linha, desta vez da direita para a esquerda.
4. As etapas 2 e 3 se repetem até que todo o tabuleiro tenha sido percorrido.

Infelizmente, Pacman não pode ignorar os comandos do usuário para fugir dos fantasmas ou pegar mais comida, mas ele pode, a qualquer momento, se aproveitar de um bug de implementação e interromper o jogo, levando consigo toda a comida que estiver carregando.

Você deve escrever um programa que determine a maior quantidade de comida que o Pacman pode levar, se escolher a melhor hora possível para sair. Note que o jogador também tem a opção de não sair antes do final do jogo.

Entrada

A primeira linha contém um inteiro N , o tamanho do tabuleiro do jogo, que é quadrado. Cada uma das N linhas seguintes contém N caracteres, que podem ser (aspas para melhor clareza):

- "." um espaço vazio;
- "o" uma comida;
- "A" um fantasma.

Saída

Seu programa deve produzir uma única linha contendo um único inteiro, a quantidade máxima de comida que o Pacman pode levar para casa.

Restrições

- $2 \leq N \leq 100$

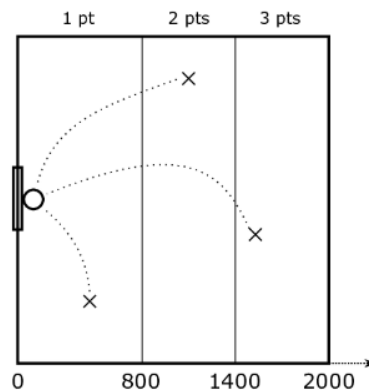
- Não há um fantasma e uma comida na mesma posição.
- Não há fantasma nem comida na posição inicial do Pacman (ou seja, o primeiro caractere da primeira linha do tabuleiro é ".").

Exemplos

Entrada 5 .ooo. ..ooA ..Aoo Aoooo ..ooo	Saída 6
Entrada 3 .o. oAA ooo	Saída 4

Basquete de Robôs

A organização da OIBR, Olimpíada Internacional de Basquete de Robô, está começando a ter problemas com dois times: os *Bit Warriors* e os *Byte Bulls*. É que os robôs desses times acertam quase todos os lançamentos, de qualquer posição na quadra! Pensando bem, o jogo de basquete ficaria mesmo sem graça se jogadores conseguissem acertar qualquer lançamento, não é mesmo? Uma das medidas que a OIBR está implantando é uma nova pontuação para os lançamentos, de acordo com a distância do robô para o início da quadra. A quadra tem 2000 centímetros de comprimento, como na figura.



Dada a distância D do robô até o início da quadra, onde está a cesta, a regra é a seguinte:

- Se $D \leq 800$, a cesta vale 1 ponto;
- Se $800 < D \leq 1400$, a cesta vale 2 pontos;
- Se $1400 < D \leq 2000$, a cesta vale 3 pontos.

A organização da OIBR precisa de ajuda para automatizar o placar do jogo. Dado o valor da distância D , você deve escrever um programa para calcular o número de pontos do lançamento.

Entrada

A primeira e única linha da entrada contém um inteiro D indicando a distância do robô para o início da quadra, em centímetros, no momento do lançamento.

Saída

Seu programa deve produzir uma única linha, contendo um inteiro, 1, 2 ou 3, indicando a pontuação do lançamento.

Restrições

- $0 \leq D \leq 2000$

Exemplos

Entrada 1720	Saída 3
Entrada 250	Saída 1
Entrada 1400	Saída 2

Drone de Entrega

A loja do Pará, especializada em vendas pela internet, está desenvolvendo drones para entrega de caixas com as compras dos clientes. Cada caixa tem a forma de um paralelepípedo reto retângulo (ou seja, no formato de um tijolo).

O drone entregará uma caixa de cada vez, e colocará a caixa diretamente dentro da casa do cliente, através de uma janela. Todas as janelas dos clientes têm o formato retangular e estão sempre totalmente abertas. O drone tem um aplicativo de visão computacional que calcula exatamente as dimensões H e L da janela. O drone consegue colocar a caixa através da janela somente quando uma das faces da caixa está paralela à janela, mas consegue virar e rotacionar a caixa antes de passá-la pela janela.

O aplicativo de controle do drone está quase pronto, mas falta um pequeno detalhe: um programa que, dadas as dimensões da maior janela do cliente e as dimensões da caixa que deve ser entregue, determine se o drone vai ser capaz de entregar a compra (pela janela) ou se a compra terá que ser entregue por meios normais.

Entrada

A entrada é composta por cinco linhas, cada uma contendo um número inteiro. As três primeiras linhas contêm os valores A, B, C, indicando as três dimensões da caixa, em centímetros. As duas últimas linhas contêm os valores H e L, indicando a altura e a largura da janela, em centímetros.

Saída

Seu programa deve escrever uma única linha, contendo apenas a letra S se a caixa passa pela janela e apenas a letra N em caso contrário.

Restrições

- $1 \leq A, B, C \leq 100$
- $1 \leq H, L \leq 100$

Exemplos

Entrada 30 50 80 80 60	Saída S
Entrada 75 100 50 100 30	Saída N

Entrada	Saída
20	S
22	
5	
20	
10	

Capital

O governo do estado de Queensland está com problemas sérios de trânsito na capital Brisbane, onde estão os prédios administrativos. Para desafogar o trânsito, o prefeito de Brisbane e o governador de Queensland decidiram que uma nova capital administrativa deve ser construída em uma área fora de Brisbane. Para projetar a nova capital, o renomado arquiteto minimalista Joe Bloggs foi contratado.

Bloggs foi informado de que o terreno destinado à nova capital ainda não foi demarcado, mas será retangular. Além disso, a cidade deverá ser dividida em quatro zonas, uma delas destinada a uma reserva ambiental e cada uma das outras três receberá os novos prédios de cada um dos três poderes (Executivo, Legislativo e Judiciário). Em um arroubo de criatividade, Bloggs decidiu que duas avenidas, perpendiculares entre si, cada uma paralela a dois dos lados do terreno retangular, dividirão a capital nas quatro zonas.

Bloggs recebeu do governo as áreas de cada uma das zonas e, após muito esforço, encontrou um retângulo que pode ser dividido conforme seus planos e de forma a respeitar as áreas delimitadas. No entanto, a Fundação de Conservação dos Cangurus determinou que a área destinada à reserva ambiental era muito pequena, o que obrigou o governo a alterar as áreas das quatro zonas. Após receber as novas medidas, Bloggs tentou encontrar um novo retângulo que viabilizasse seu projeto, porém sem sucesso. Cansado de fazer testes, ele pensou que talvez tenha que abandonar sua brilhante ideia. Por isso, ele pediu para você escrever um programa que, dadas as áreas das quatro zonas, determine se ele poderá ou não manter seu projeto (ou seja, se existe um retângulo que possa ser dividido por duas retas perpendiculares, cada uma paralela a dois dos lados do retângulo, tal que as quatro áreas formadas obedeçam às exigências do governo).

Entrada

A entrada consiste de uma única linha contendo quatro inteiros A_1 , A_2 , A_3 , A_4 , indicando a área de cada uma das zonas.

Saída

Imprima uma única linha contendo um único caractere: "S" se Bloggs pode preservar seu projeto e "N" caso contrário.

Restrições

- $1 \leq A_i \leq 10^4$

Exemplos

Entrada 1 2 4 8	Saída S
Entrada 1 2 3 4	Saída N
Entrada 15 14 6 35	Saída S

Cobra Coral

O professor Rui está desenvolvendo um sistema automático para identificar se uma cobra é uma coral verdadeira ou uma falsa coral. A cobra coral verdadeira é venenosa e os anéis coloridos no seu corpo seguem o padrão ...BVBPBVBPBVBP..., onde B,V e P representam as cores branco, vermelho e preto, respectivamente. Já a falsa coral não é venenosa e os anéis seguem o padrão ...BVPBVBPBVBP....

O problema é que os sensores do sistema do professor Rui produzem apenas uma sequência de quatro números representando um pedaço do padrão de cores. Só que ele não sabe qual número representa qual cor. Mas, por exemplo, se a sequência for 5 3 9 3, podemos dizer com certeza que é uma coral verdadeira, mesmo sem saber qual número representa qual cor! Você deve ajudar o professor Rui e escrever um programa que diga se a coral é verdadeira ou falsa.

Entrada

A entrada consiste de apenas uma linha, contendo quatro números inteiros.

Saída

Seu programa deve imprimir na saída uma linha com a letra "V" se a coral for verdadeira ou com a letra "F", caso seja falsa.

Restrições

Os quatro números têm valores entre 1 e 9, inclusive, e a sequência sempre representa uma coral verdadeira, ou uma coral falsa.

Exemplos

Entrada 5 3 9 3	Saída V
Entrada 7 1 4 7	Saída F
Entrada 6 2 6 8	Saída V

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