

CADERNO DE QUESTÕES

A competição terá uma duração de 4 horas (13hs às 17hs) e é composta de uma série de problemas listados neste caderno.

- É permitido que as equipes utilizem material impresso para consulta (livros, apostilas, anotações).
- Não é permitido o acesso a Internet para consulta de material, bem como o acesso qualquer conteúdo armazenado em meio magnético ou eletrônico.
- É vetado aos alunos se comunicarem com os membros de outras equipes. Apenas 1 computador é alocado para cada equipe durante a competição.
- Leia atentamente os enunciados.
- O caderno de problemas possui problemas fáceis, moderados e difíceis. Se um problema parecer difícil, considere abordar outro problema. Entretanto, alguns problemas podem parecer difíceis, e outros podem parecer fáceis.
- Certifique-se de que o problema realmente foi entendido, antes de partir para a implementação.
- Certifique-se de que o programa está correto, antes de submetê-lo.

Boa sorte!

Patrocinadores:



Problema A

Jogo do Quadrado

Arquivo fonte: quadrado.[c,cpp,java]

O "jogo do quadrado" é um jogo muito popular hoje em dia! O jogo é muito simples: é dada um retângulo de **N** linhas e **M** colunas contendo números inteiros não negativos. A imagem a seguir mostra um retângulo com 3 linhas e 4 colunas.

3	4	0	3
0	2	3	1
4	2	1	0

Também é dado um inteiro **S**. Você deve escolher algum quadrado com **S** linhas e **S** colunas contido inteiramente dentro do retângulo. Sua pontuação é dada pelo produto de todos os inteiros dentro do quadrado que você escolheu. Por exemplo, se **S=2** e você escolheu o quadrado mostrado com sombreamento da matriz acima, sua pontuação será igual a $2 \times 3 \times 2 \times 1 = 12$.

Você percebeu que, dependendo do quadrado que você escolher, sua pontuação pode ser igual a zero. São dados um retângulo e uma lista de consultas. Para cada consulta, é dado um inteiro **S** e você deve determinar se é possível escolher algum quadrado **SxS** de tal forma que sua pontuação não será igual a zero.

Entrada

A primeira linha contém dois inteiros **N** e **M** ($1 \leq N, M \leq 200$) indicando o número de linhas e de colunas do retângulo. As próximas **N** linhas contém **M** inteiros cada, descrevendo o retângulo. Cada inteiro no retângulo não é maior que 10^9 .

A próxima linha contém um inteiro **Q** ($1 \leq Q \leq 200$) indicando o número de consultas. Cada uma das próximas **Q** linhas descreve uma consulta. Cada linha contém um inteiro **S** ($1 \leq S \leq \min(N, M)$) indicando o comprimento do lado do quadrado que você deve escolher.

Saída

Para cada consulta, imprima uma linha contendo **SIM** se é possível escolher um quadrado tal que sua pontuação não será igual a zero, ou **NAO** caso contrário.

Exemplo de Entrada	Exemplo de Saída
3 4	SIM
3 4 0 3	NAO
0 2 3 1	SIM
4 2 1 0	
3	
2	
3	
1	

Problema B

Vírus, e agora Diego?

Arquivo fonte: virus.[c,cpp,java]

Diego está cursando o último ano de Análise e Desenvolvimento de Sistemas e se preparando para ingressar no mercado de trabalho. Para conclusão de sua graduação é necessária entrega do Relatório Técnico do desenvolvimento realizado. Diego concluiu o trabalho e está preparado para realizar a impressão, porém ele percebeu que sua impressora foi infectada por um vírus denominado “nic” e está imprimindo de forma incorreta. Depois de olhar para várias páginas impressas por um tempo, Diego percebe que ele está imprimindo cada linha de dentro para fora. Em outras palavras, a metade esquerda de cada linha está sendo impressa a partir do meio da página até a margem esquerda. Do mesmo modo, a metade direita de cada linha está sendo impressa à partir da margem direita e prosseguindo em direção ao centro da página.

Por exemplo a linha:

O MODELO DE ENGENHARIA DE SOFTWARE

Está sendo impressa como:

NEGNE ED OLEDOM OERAWTFOS ED AIRAH

Da mesma forma, a linha "CLASSE" está sendo impressa incorretamente como "ALCESS". Sua tarefa é decifrar a string a partir da forma como ela foi impressa para a sua forma original. Você pode assumir que cada linha conterá um número par de caracteres.

Sua tarefa nesse problema é ajudar o Diego a imprimir o texto original para entrega e conclusão do curso.

Entrada

A entrada contém vários casos de teste. A primeira linha de entrada contém um inteiro **N** que indica a quantidade de casos de teste. Seguem **N** linhas, cada uma com uma frase com no mínimo 2 e no máximo 100 caracteres de letras maiúsculas e espaços que deverá ser decifrada à partir da forma impressa para a sua forma original, conforme especificação acima.

Saída

Para cada linha de entrada deverá ser impressa uma linha de saída com a frase decifrada, conforme a especificação acima.

Exemplo de Entrada	Exemplo de Saída
3 M ADUJA AIGOLONCET ASERODEDNEERPME SOTIU D SOSSECORPSEAWTFOS E ED OIFASEDOACAMARGORP	A TECNOLOGIA AJUDA MUITOS EMPREENDEDORES PROCESSOS DE SOFTWARES DESAFIO DE PROGRAMACAO

Problema C

Variância e Acima da Média

Arquivo fonte: *variancia.[c,cpp,java]*

Os professores de programação do curso de Gestão da Tecnologia da Informação perceberam que os alunos têm sempre a expectativa de serem acima da média, sendo assim eles decidiram acirrar essa disputa de forma sadia com a intenção de evolução dos alunos dentro da programação. Embora essa disputa seja importante para as disciplinas os professores têm a preocupação que ocorra a dispersão entre as notas. Quanto menor a dispersão entre os alunos, melhor aproveitamento teve a sala.

Você deve construir um programa para calcular a média aritmética, a variância e indicar a porcentagem de alunos que estavam acima da média.

$$v = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

Onde:

v = variância

n = número de elementos

x_i = nota do aluno i

\bar{x} = média dos alunos

Exemplo:

Dadas 5 notas: 50 50 70 80 100

Média aritmética: $(50 + 50 + 70 + 80 + 100) / 5 = 70$

Variância: $((50-70)^2 + (50-70)^2 + (70-70)^2 + (80-70)^2 + (100-70)^2) / 5 = 360$

Porcentagem de alunos acima da média: 40%

Entrada

A entrada contém muitos casos de teste. A primeira linha da entrada contém um inteiro **C**, indicando o número de casos de teste. Seguem **C** casos de teste ou instâncias. O terminador de entrada é encerrado caso seja digitado um valor igual a zero (**C=0**). Cada caso de teste inicia com um inteiro **N**, que é o número de pessoas de uma turma ($1 \leq N \leq 1000$). Seguem **N** inteiros, separados por espaços, cada um indicando a média final (um inteiro entre 0 e 100) de cada um dos estudantes desta turma.

Saída

Para cada caso de teste imprima uma linha indicando o caso seguido de um número sequencial, conforme o exemplo: "CASO #I:", onde I indica o número sequencial dos casos. Na próxima linha deverá ser informado o percentual de estudantes que estão acima da média da turma, com o valor arredondado e com 2 casas decimais, separado por um espaço que indique a variância, também com duas casas decimais.

Uma linha em branco deve separar a saída de cada caso.

Exemplo de Entrada	Exemplo de Saída
4 5 50 50 70 80 100 7 100 95 90 80 70 60 50 3 70 90 80 3 70 90 81 0	CASO #1: 40.00% 360.00 CASO #2: 57.14% 298.98 CASO #3: 33.33% 66.67 CASO #4: 66.67% 66.89

Problema D

Lenda da Matriz

Arquivo fonte: matriz.[c,cpp,java]

As matrizes surgiram da necessidade de resolver problemas, que envolviam mensuração de terras, agricultura, impostos e etc., os quais resultavam em sistemas de equações de 1º grau.

Hoje em dia, as matrizes têm uma importância muito significativa no campo das aplicações em Matemática e Computação Gráfica. Também são muito utilizadas em nosso cotidiano, por exemplo, na organização de dados, calendário, ficha de aposta de loteria e até a tela do computador que você observa agora é formada por pixels gerado por uma matriz.

Existe também uma lenda que os alunos de exata têm facilidade em trabalhar com dados disposto em matrizes. Pensado nisso, você foi convidado para trabalhar com este teste, criando um algoritmo para trabalhar com dados de uma matriz e por esta lenda a prova.

Entrada

Dada uma matriz quadrada da ordem de 'N' onde 'N' é um número digitado pelo usuário e $2 \leq 'N' \leq 10$. Sendo os campos desta matriz preenchidos com números sequenciais a partir de um número 'M' também informado pelo usuário, onde $1 \leq 'M' \leq 1000$.

A entrada dos dados ocorrerá em uma única linha sendo na ordem 'N' e 'M'. O código é finalizado quando o usuário digitar 0 0 (zero zero) referente aos valores 'N' e 'M'.

Saída

A saída dos dados também ocorrerá em uma única linha, devendo conter a diagonal principal da matriz e a quantidade de números primos que a diagonal principal possui.

Exemplo:

Por exemplo, se os dados correspondentes a 'N' e 'M' forem 5 e 11, o algoritmo criará a seguinte matriz (atenção, esta matriz não será impressa):

Matriz

11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35

E terá como saída a diagonal principal (11 17 23 29 35) e a quantidade de números primos (4)

Exemplo de Entrada	Exemplo de Saída
2 2 5 11 0 0	2 5 2 11 17 23 29 35 4

Problema E

Quadripartidos

Arquivo fonte: quadripartido.[c,cpp,java]

Malba Tahan, em seu clássico "O Homem Que Calculava", conta uma fábula de superstição envolvendo os números quadripartidos. Mal sabia ele que séculos antes, na antiga civilização Tcheca, a superstição envolvendo os números quadripartidos já se fazia presente. Na antiguidade, uma importante comunidade que vivia nos arredores de Neratovice, utilizava as propriedades dos números quadripartidos para prever o futuro, batizar as crianças e até mesmo para escolher os seus líderes.

Um número inteiro n é quadripartido se existe alguma divisão desse número em quatro parcelas inteiras ($p_1 + p_2 + p_3 + p_4 = n$) e um operador mágico (m) de modo que a primeira parcela somada ao operador mágico, a segunda diminuída dele, a terceira multiplicada por ele e a quarta dividida por ele deem o mesmo resultado ($p_1 + m = p_2 - m = p_3 * m = p_4 / m$).

Assim, 128 é quadripartido, porque podemos dividir 128 em 4 parcelas (31, 33, 32 e 32) de modo que existe um operador mágico (no caso, 1) que faz com que $p_1 + m$, $p_2 - m$, $p_3 * m$ e p_4 / m sejam iguais. De fato: $31 + 1 = 33 - 1 = 32 * 1 = 32 / 1 = 32$.

Um grupo de pesquisadores de Praga está reconstruindo o passado de Neratovice, e pediu a sua ajuda. Eles querem que você faça um programa que identifique quando um número é ou não quadripartido e qual é o seu operador mágico associado.

Entrada

Cada linha da entrada contém um inteiro n ($0 \leq n \leq 500000$) que seu programa deverá analisar e classificar em quadripartido ou não. O valor $n = 0$ corresponde ao final do arquivo de entrada e não deve ser processado.

Saída

Para cada valor da entrada, seu programa deve imprimir um identificador "**INSTANCIA #h**", em que h é um número inteiro, sequencial e crescente a partir de 1. Na linha seguinte, separados por um espaço em branco, os cinco números que comprovam a condição de quadripartido, quando n for quadripartido. Siga a ordem: **m p₁ p₂ p₃p₄**.

Se n não for quadripartido, seu programa deve imprimir a mensagem "*". No primeiro caso, é possível que exista mais de uma sequência que atenda às condições estabelecidas. Se isto ocorrer, seu programa deverá escolher a que apresentar o maior valor possível para m .

Uma linha em branco deve separar a saída de cada instância.

Exemplo de Entrada	Exemplo de Saída
128 1 8 1000 1001 0	INSTANCIA #1: 7 7 21 2 98 INSTANCIA #2: * INSTANCIA #3: 1 1 3 2 2 INSTANCIA #4: 9 81 99 10 810 INSTANCIA #5: *

Problema F

Fila dos Brindes

Arquivo fonte: brindes.[c,cpp,java]

A Fatec Guaratinguetá está organizando um evento com várias palestras na área de Tecnologia da Informação. Nas palestras serão distribuídos brindes a todos os alunos, por isso essa hora é a mais aguardada pela grande maioria dos alunos. Não só porque os brindes são interessantes, mas sim porque os melhores brindes serão dados para os primeiros alunos da fila.

Quando foi indicado o início da distribuição dos brindes, todos os alunos saíram correndo para chegar às primeiras posições da fila, tanta era o desejo. Um de seus professores notou, porém, que havia ali uma oportunidade.

Utilizando um sistema de recompensa, sua professora de matemática disse que a ordem da fila para o recebimento do brinde será dada não pela ordem de chegada, mas sim pelas notas obtidas em sala de aula. Assim, aqueles com maior nota serão os primeiros da fila.

Sua tarefa é simples: dada a ordem de chegada dos alunos, e as suas respectivas notas na matéria de matemática, reordene a fila de acordo com as notas de matemática, e diga quantos alunos não precisaram trocar de lugar nessa reordenação.

Entrada

A primeira linha contém um inteiro **N**, indicando o número de casos de teste a seguir.

Cada caso de teste inicia com um inteiro **M** ($1 \leq M \leq 1000$), indicando o número de alunos. Em seguida haverá **M** inteiros distintos **P_i** ($1 \leq P_i \leq 1000$), onde o **i**-ésimo inteiro indica a nota do **i**-ésimo aluno.

Os inteiros acima são dados em ordem de chegada, ou seja, o primeiro inteiro diz respeito ao primeiro aluno a chegar na fila, o segundo inteiro diz respeito ao segundo aluno, e assim sucessivamente.

Saída

Para cada caso de teste imprima uma linha, contendo um inteiro, indicando o número de alunos que não precisaram trocar de lugar mesmo após a fila ser reordenada.

Exemplo de Entrada	Exemplo de Saída
4	1
3	0
100 80 90	4
4	3
100 120 30 50	
4	
100 90 30 25	
5	
60 80 40 30 30	

Problema G

Números Romanos

Arquivo fonte: romanos.[c,cpp,java]

A Fatec Guaratinguetá decidiu adotar em suas monografias a numeração romana. Porém, muitos alunos não se lembram de como utilizar esse tipo de numeração e decidiu perguntar aos desenvolvedores da faculdade se eles poderiam criar um programa que dada uma entrada de um valor numérico (arábico) inteiro o programa deveria mostrar o seu equivalente na numeração romana.

Lembre-se que:

I: 1	X: 10	C: 100	M: 1000
V: 5	L: 50	D: 500	

Entrada

A entrada é um número inteiro positivo **N** ($0 < N < 1000$). O programa será encerrado se for digitado um valor para $N=0$ (zero).

Saída

A saída é o número **N** escrito na numeração romana em uma única linha. Use sempre letras maiúsculas.

Exemplos de Entrada	Exemplos de Saída
666	DCLXVI
83	LXXXIII
999	CMXCIX
236	CCXXXVI
0	

Problema H

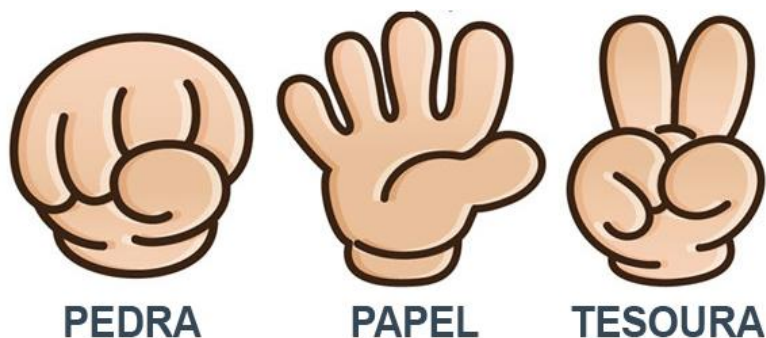
Jokenpô!

Arquivo fonte: jokenpo.[c,cpp,java]

Jokenpô, como é conhecido no Brasil, ou pedra - papel - tesoura, é um jogo de mãos recreativo e simples para duas ou mais pessoas. O jogo é frequentemente empregado como método de seleção (como na escolha de equipes para a prática desportiva, por exemplo), assim como lançar moedas, jogar dados, entre outros. No entanto, diferentemente desses métodos que se baseiam exclusivamente em sorte, pedra-papel-tesoura pode ser jogado com um pouco de habilidade. Principalmente se o jogo se estender por vários turnos com o mesmo jogador, este pode reconhecer e explorar a lógica do comportamento do adversário (perceber e anteceder as jogadas do adversário).

As regras do jogo proposto são:

1. a tesoura corta o papel;
2. o papel embrulha a pedra;
3. a pedra quebra a tesoura;



Entrada

A entrada consiste em uma série de casos de teste. A primeira linha contém um inteiro positivo T ($T \leq 100$), que representa o número de casos de teste. Cada caso de teste é representado por uma linha da entrada, contendo as escolhas de jogador1 e jogador2, respectivamente, separadas por um espaço em branco. As escolhas possíveis são as personagens do jogo: pedra, papel e tesoura.

Saída

Para cada caso de teste deverá ser impressa a mensagem "CASO #i: R", onde **i** é o número do caso de teste (cuja contagem se inicia no número um) e **R** é um dos três resultados possíveis: "JOGADOR1", "JOGADOR2", ou "EMPATE".

Exemplos de Entrada	Exemplos de Saída
3 PAPEL PEDRA PAPEL TESOURA PEDRA PEDRA	CASO #1: JOGADOR1 CASO #2: JOGADOR2 CASO #3: EMPATE