



7º Hackathon Fulltime

FATEC Garça

(17 de agosto de 2024)

Caderno de Problemas

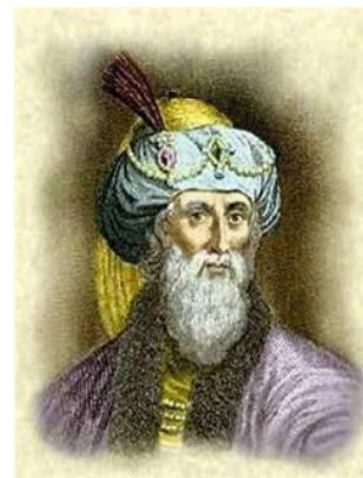
Informações Gerais

- Este caderno contém 10 problemas – A a J, com páginas numeradas de 01 a 11. Verifique se o caderno está completo.
- A competição possui duração de 4 horas (início as 13:00h término as 17:00h);
- Durante a competição é permitida aos competidores a consulta a material próprio impresso em papel (livros, apostilas, anotações);
- NÃO é permitido acesso a conteúdo da Internet e nem conteúdos disponíveis em qualquer meio magnético;
- É vedada a comunicação entre os competidores durante a competição, bem como a troca de material de consulta entre eles;
- Cada competidor terá acesso a 1 computador dotado do ambiente de submissão de programas beecrowd, dos compiladores, link-editores e IDEs requeridos pelas linguagens de programação permitidas;
- Não é permitido o uso de notebooks ou outro tipo de computador ou assistente pessoal;
- Os problemas têm o mesmo valor na correção;
- A correção é automatizada, portanto siga atentamente as exigências do problema quanto ao formato da entrada e saída de seu programa;
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), todos os dados devem ser lidos da entrada padrão e escritos na saída padrão. Nem utilize qualquer rotina para limpar a tela ou posicionar o cursor;
- Preste muita atenção na linguagem escolhida;
- As soluções serão testadas com outras entradas além das apresentadas como exemplo nos problemas;

Problema A

A Lenda de Flavius Josephus

O problema de Josephus é assim conhecido por causa da lenda de Flavius Josephus, um historiador judeu que viveu no século 1. Segundo o relato de Josephus do cerco de Yodfat, ele e seus companheiros (40 soldados) foram presos em uma caverna, cuja saída foi bloqueada pelos romanos. Ele e seus companheiros preferiram suicidar-se a serem capturados, e decidiram que iriam formar um círculo e começar a matar-se pulando de três em três. Josephus afirma que, por sorte ou talvez pela mão de Deus, ele permaneceu por último e preferiu entregar-se aos romanos a suicidar-se.



Entrada

Haverá **NC** ($1 \leq \text{NC} \leq 30$) casos de teste. Em cada caso de teste de entrada haverá um par de números inteiros positivos **n** ($1 \leq n \leq 10000$) e **k** ($1 \leq k \leq 1000$). O número **n** representa a quantidade de pessoas no círculo, numeradas de 1 até **n**. O número **k** representa o tamanho do salto de um homem até o próximo homem que será morto.

Segue um exemplo com 5 homens e um salto = 2.



Neste exemplo o elemento que restará após as eliminações é 3.

Saída

Para cada caso de teste de entrada será apresentada uma linha de saída no seguinte formato: *Case n: m* tendo sempre um espaço antes do **n** e do **m**.

Exemplos

Entrada	Saída
3	Case 1: 3
5 2	Case 2: 1
6 3	Case 3: 25
1234 233	

Problema B

Balanço de Parênteses I

Dada uma expressão qualquer com parênteses, indique se a quantidade de parênteses está correta ou não, sem levar em conta o restante da expressão. Por exemplo:

$a + (b * c) - 2 - a$ está correto

$a + b * (2 - c) - 2 + a) * 2$ está correto

enquanto

$(a * b - (2 + c)$ está incorreto

$2 * (3 - a))$ está incorreto

$) 3 + b * (2 - c) ($ está incorreto

Ou seja, todo parênteses que fecha deve ter um outro parênteses que abre correspondente e não pode haver parênteses que fecha sem um prévio parênteses que abre e a quantidade total de parênteses que abre e fecha deve ser igual.

Entrada

Como entrada, haverá **N** expressões ($1 \leq N \leq 10000$), cada uma delas com até 1000 caracteres.

Obs.: percebam que no exemplo não tem a entrada de N, são N expressões, ou seja, enquanto for diferente de fim de arquivo.

Saída

O arquivo de saída deverá ter a quantidade de linhas correspondente ao arquivo de entrada, cada uma delas contendo as palavras **correct** ou **incorrect** de acordo com as regras acima fornecidas.

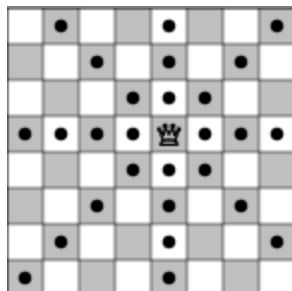
Exemplos

Entrada	Saída
$a + (b * c) - 2 - a$	correct
$(a + b * (2 - c) - 2 + a) * 2$	correct
$(a * b - (2 + c)$	incorrect
$2 * (3 - a))$	incorrect
$) 3 + b * (2 - c) ($	incorrect

Problema C

Dama

O jogo de xadrez possui várias peças com movimentos curiosos: uma delas é a *dama*, que pode se mover qualquer quantidade de casas na mesma linha, na mesma coluna, ou em uma das duas diagonais, conforme exemplifica a figura abaixo:



O grande mestre de xadrez Kary Gasparov inventou um novo tipo de problema de xadrez: dada a posição de uma dama em um tabuleiro de xadrez vazio (ou seja, um tabuleiro 8×8 , com 64 casas), de quantos movimentos, no mínimo, ela precisa para chegar em outra casa do tabuleiro?

Kary achou a solução para alguns desses problemas, mas teve dificuldade com outros, e por isso pediu que você escrevesse um programa que resolve esse tipo de problema.

Entrada

A entrada contém vários casos de teste. A primeira e única linha de cada caso de teste contém quatro inteiros X_1, Y_1, X_2 e Y_2 ($1 \leq X_1, Y_1, X_2, Y_2 \leq 8$). A dama começa na casa de coordenadas (X_1, Y_1) , e a casa de destino é a casa de coordenadas (X_2, Y_2) . No tabuleiro, as colunas são numeradas da esquerda para a direita de 1 a 8 e as linhas de cima para baixo também de 1 a 8. As coordenadas de uma casa na linha X e coluna Y são (X, Y) .

O final da entrada é indicado por uma linha contendo quatro zeros.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha na saída, contendo um número inteiro, indicando o menor número de movimentos necessários para a dama chegar em sua casa de destino.

Exemplos

Entrada	Saída
4 4 6 2	1
3 5 3 5	0
5 5 4 3	2
0 0 0 0	

Problema D

Flores Florescem da França

Fiona sempre amou poesia, e recentemente descobriu uma forma poética fascinante. *Tautogramas* são um caso especial de aliteração, que é a ocorrência da mesma letra no início de palavras adjacentes. Em particular, uma sentença é um tautograma se todas suas palavras começam com a mesma letra.

Por exemplo, as seguintes sentenças são tautogramas:

- Flowers Flourish from France
- Sam Simmonds speaks softly
- Peter pIckEd pePPers
- truly tautograms triumph

Fiona quer deslumbrar seu namorado com uma carta romântica repleta desse tipo de sentenças. Por favor, ajude Fiona a verificar se cada sentença que ela escreveu é um tautograma ou não.

Entrada

Cada caso de teste é dado em uma única linha que contém uma frase. Uma frase consiste de uma sequência de no máximo 50 palavras separadas por espaços simples. Uma palavra é uma sequência de no máximo 20 letras do alfabeto inglês (maiúsculas e minúsculas). Uma palavra contém ao menos uma letra e uma sentença contém ao menos uma palavra.

O último caso de teste é seguido de uma linha contendo apenas o caractere '*' (asterisco).

Saída

Para cada caso de teste, imprima uma única linha contendo o caractere 'Y' se a sentença é um tautograma, ou um 'N' caso contrário.

Exemplos

Entrada	Saída
Flowers Flourish from France	Y
Sam Simmonds speaks softly	Y
Peter pIckEd pePPers	Y
truly tautograms triumph	Y
this is NOT a tautogram	N
*	

Problema E

Trigo no Tabuleiro

Uma rainha requisitou os serviços de um monge e disse-lhe que pagaria qualquer preço. O monge, necessitando de alimentos, perguntou a rainha se o pagamento poderia ser feito em grãos de trigo dispostos em um tabuleiro de damas, de forma que o primeiro quadrado tivesse apenas um grão, e os quadrados subsequentes, o dobro do quadrado anterior. A rainha considerou o pagamento barato e pediu que o serviço fosse executado, porém, um dos cavaleiros que estava presente e entendia um pouco de matemática alertou-a que seria impossível executar o pagamento, pois a quantidade de grão seria muito alta. Curiosa, a rainha solicitou então a este cavaleiro que era bom em cálculo, que fizesse um programa que recebesse como entrada o número de quadrados a serem usados em um tabuleiro de damas e apresentasse a quantidade de kg de trigo correspondente, sabendo que cada 12 grãos do cereal correspondem a uma grama. Finalmente, o cálculo da quantidade deverá caber em um valor inteiro de 64 bits sem sinal.

Entrada

A primeira linha de entrada contém um único inteiro N ($1 \leq N \leq 100$), indicando o número de casos de teste. Cada caso de teste contém um único inteiro X ($1 \leq X \leq 64$), indicando o número de casas do tabuleiro que serão utilizadas.

Saída

Para cada caso de teste, imprima a quantidade de kgs de trigo que o monge esperava receber.

Exemplos

Entrada	Saída
3	0 kg
7	43 kg
19	1 kg
14	

Problema F

O Bravo Guerreiro Hashmat

Hashmat é um guerreiro cujo seu grupo de soldados se move de um local a outro para lutar contra os seus oponentes. Antes da luta ele apenas calcula uma coisa. A diferença entre a quantidade de soldados que possui e a quantidade de soldados oponentes. A partir desta diferença ele decide se vai ou não lutar. Às vezes Hashmat tem mais soldados do que o seu oponente, mas na maioria das vezes não.

Entrada

A entrada contém dois números inteiros em cada linha. Estes dois números denotam respectivamente a quantidade de soldados do exército de Hashmat e do seu oponente. Nenhum número de entrada é maior do que 2^{32} . A entrada termina com fim de arquivo (EOF).

Saída

Para cada linha de entrada imprima a diferença entre o número de soldados de Hashmat e do seu oponente. Cada saída deve ser impressa em uma linha separada.

Exemplos

Entrada	Saída
10 12	2
10 14	4
100 200	100

Problema G

Getline One

Mangojata está aprendendo programação. Ela acha tudo muito fácil, muito simples. Ela está prestes a fazer um pequeno programa que leia o nome dos seus amigos e a distância de sua casa até cada um deles. Desta forma, ela quer simplesmente calcular qual é a distância média que deve ser percorrida para chegar na casa de qualquer um de seus amigos (em metros). Porém Aristoclenes, que é um programador mais experiente, lhe alertou que às vezes o que parece muito simples tem lá seus detalhes, dependendo da linguagem que é utilizada para implementação.

Entrada

A entrada contém vários casos de teste e termina com EOF (Fim de Arquivo). Cada caso de teste consiste de duas linhas de entrada. A primeira linha contém o nome de um amigo de Mangojata e a segunda linha contém um valor inteiro que indica a distância aproximada da casa deste amigo até a casa de Mangojata.

Saída

A saída deve ser um único valor com uma casa decimal (utilize uma variável de dupla precisão - double) indicando a distância média entre a casa de Mangojata e de seus amigos, conforme exemplo abaixo.

Exemplos

Entrada	Saída
Juca Pivara 410	160.7
Pedro Medario 12	
Marta Mandua 60	

Problema H

Plano de Dieta

O doutor deu a você a sua dieta, na qual cada caractere corresponde a algum alimento que você deveria comer. Você também sabe o que você tem comido no café da manhã e no almoço, nos quais cada caractere corresponde a um tipo de alimento que você deveria ter comido aquele dia. Você decidiu que irá comer todo o restante de sua dieta durante o jantar, e você quer imprimi-la como uma String (ordenada em ordem alfabética). Se você trapaceou de algum modo (ou por comer muito de tipo de alimento, ou por comer algum alimento que não está no plano de dieta), você deveria imprimir a cadeia "CHEATER" (significa trapaceiro), sem as aspas.

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. Cada caso de teste é composto por três linhas, cada uma delas com uma string com até 26 caracteres de 'A'-'Z' ou vazia, representando respectivamente os alimentos da dieta, do café da manhã e do almoço.

Saída

Para cada caso de teste imprima uma string que representa os alimentos que você deveria consumir no jantar, ou "CHEATER" caso você tenha trapaceado na sua dieta.

Exemplos

Entrada	Saída
5 ABCD AB C ABEDCS EDSMB MSD A IWANTSODER SOW RAT	D ABCDES CHEATER DEIN

Problema I

Tudo o que você precisa é amor

*"All you need is love. All you need is love.
All you need is love, love... love is all you need."
The Beatles*

Foi inventado um novo dispositivo poderoso pela *Beautifull Internacional Machines Corporation* chamado de "Máquina do amor!". Dada uma string feita de dígitos binários, a máquina do amor responde se isto é feito somente de amor, ou seja, se tudo o que você irá precisar para construir aquela string for somente amor. A definição de amor para a Máquina do amor é outra string de dígitos binários, fornecida por um operador humano. Vamos supor que nós temos uma string L que representa "love" e forneçamos uma string S para a máquina do amor. Diremos então que tudo o que você precisa é amor para construir S se pudermos repetidamente subtrair L de S até que sobre apenas L . A subtração definida aqui é a mesma subtração aritmética binária na base 2. Por definição é fácil de ver que $L > S$ (em binário), então S não é feito de amor. Se $S = L$ então S é obviamente feito de amor.

Por exemplo, suponha $S = "11011"$ e $L = "11"$. Se repetidamente subtrairmos L de S , obteremos: 11011, 11000, 10101, 10010, 1111, 1100, 1001, 110, 11. Portanto, dado este L , tudo o que você necessita é amor para construir S . Devido a algumas limitações da Máquina do Amor, não será possível lidar com strings com zero à esquerda. Por exemplo "0010101", "01110101", "011111" etc. são string Inválidas. Strings que contenham apenas um dígito também são strings inválidas (isto é outra limitação).

Sua tarefa para este problema é: dadas duas strings binárias válidas, $S1$ e $S2$, veja se é possível ter uma string L válida tal que ambas, $S1$ e $S2$ possam ser feitas apenas de L (i.e. dadas duas strings válidas $S1$ e $S2$, indique se existe pelo menos uma string L válida tal que ambas $S1$ e $S2$ sejam feitas apenas de L). Por exemplo, para $S1 = 11011$ e $S2 = 11000$, nós podemos ter $L = 11$ tal que $S1$ e $S2$ são feitas ambas somente de L (como pode ser visto no exemplo abaixo).

Entrada

A primeira linha de entrada contém um valor inteiro positivo N ($N < 10000$) que indica o número de casos de teste. Então, $2*N$ linhas vem a seguir. Cada par de linhas consiste de um caso de teste. Cada par de linhas contém respectivamente $S1$ e $S2$ que serão inseridas como entrada para a máquina do amor. Nenhuma string conterá menos do que 2 ou mais do que 30 caracteres. Você pode assumir que as strings de entrada serão válidas e estarão de acordo com as regras acima.

Saída

Para cada par de strings, seu programa deve imprimir uma das seguintes mensagens:

Pair #p: All you need is love!

Pair #p: Love is not all you need!

Onde p representa o número do par de entrada (que inicia em 1). Seu programa deve imprimir a primeira mensagem no caso de existir pelo menos uma string L válida tal que ambas as strings $S1$ e $S2$ possam ser feitas somente de L . Caso contrário, imprima a segunda linha.

Exemplos

Entrada	Saída
5 11011 11000 11011 11001 111111 100 1000000000 110 1010 100	Pair #1: All you need is love! Pair #2: Love is not all you need! Pair #3: Love is not all you need! Pair #4: All you need is love! Pair #5: All you need is love!

Problema J

Jogo do Tijolo

Há uma vila em Bangladesh, onde o jogo de tijolo é muito popular. Jogo de tijolo é um jogo de equipe. Cada equipe é constituída por um número ímpar de jogadores. O número de jogadores deve ser maior do que 1, mas não pode ser superior a 10. A idade de cada jogador deve estar entre 11(inclusive) e 20(inclusive). Não há dois jogadores com a mesma idade. Há um capitão para cada equipe.

A falta de comunicação entre dois jogadores depende da sua diferença de idade, ou seja, é maior se a diferença de idade for maior. Assim, eles selecionam o capitão de uma equipe de maneira que a quantidade de jogadores desta equipe que são mais jovens e mais velhos do que ele é igual.

As idades de todos os membros da equipe são fornecidas. Você tem que determinar a idade do capitão.

Entrada

A entrada começa com um número inteiro **T** ($T \leq 100$), representando o número de casos de teste. Cada uma das próximas **T** linhas irá começar com um número inteiro **N** ($1 < N < 11$), número de membros da equipe, seguido por **N** inteiros separados por espaço representando as idades de todos os membros de uma equipe. Cada um destes **N** inteiros será entre 11 e 20(inclusive). Nota-se que, as idades serão dadas estritamente em ordem crescente ou estritamente em ordem decrescente. Nós não vamos mencionar qual está aumentando e qual está diminuindo, você tem que ser cuidadoso o suficiente para lidar com ambas as situações.

Saída

Para cada caso de teste, imprima uma linha no formato "**Case x: a**"(sem as aspas), onde **x** é o número do caso de teste e **a** é a idade do capitão.

Exemplos

Entrada	Saída
2	Case 1: 16
5 19 17 16 14 12	Case 2: 16
5 12 14 16 17 18	