

Semana 2 – Lista de Exercícios

(Esta lista é para ser feita individualmente)

Como determinado na última aula (17/02), devido ao limitação de tempo para passar o conteúdo referente ao conteúdo do framework Yesod, o projeto de Yesod, que valeria 50% da nota do curso, foi substituído por uma segunda lista de exercícios.

Cada exercício desta lista vale 1.75 pontos.

A nota desta lista será dada por:

$$L = \min(10, P)$$

Onde:

L = Nota da lista

P = Pontos obtidos na lista

Lembrando que esta lista vale 50% da nota final do curso.

Cada exercício deve ser entregue em um arquivo ExN.hs, onde N é o número do exercício (exemplo: exercício 1 será entregue no arquivo Ex1.hs). Entregue todos os arquivos dos exercícios em compactados em um único arquivo zip. O nome do arquivo zip deve ser seu nome completo.

Em caso de dúvidas pergunte ao professor ou ao monitor.

Exercício 1)

Um problema conhecido de alguns microfones é a “sibilância do S”. Ou seja, algumas vezes o som da letra S produz uma aspereza desagradável.

Palavras que contêm dois S’s seguidos são particularmente sujeitas a sibilância. Palavras como, assado, esse, assinar e classificar são exemplos que provocam sibilância do S.

Escreva um programa para determinar se uma palavra contém dois S’s seguidos.

Para fazer a verificação utilize o conceito de recursão.

Entrada

A entrada é composta por uma linha com uma única string. A string contém apenas letras minúsculas (sem espaços) entre 1 e 30 caracteres.

Saída

Seu programa deverá produzir uma única linha contendo o caractere S (maiúsculo) se a palavra contém dois S consecutivos, ou o caractere N (maiúsculo), caso contrário.

Exemplos

Entrada	Saída
assado	S
sábados	N
kiss	S

Exercício 2)

Quando você viaja para Hong Kong, a primeira preocupação é ir à casa de câmbio, pois em Hong Kong não são aceitas moedas estrangeiras.

Escreva um programa que dada a taxa de câmbio de diferentes moedas e o valor de Dólares de Hong Kong (HKD) que você precisa, calcule o valor correspondente em cada moeda estrangeira.

Entrada

A entrada consiste em $N+2$ linhas. A primeira linha contém um inteiro N ($1 \leq N \leq 15$), que representa o número de moedas estrangeiras.

Nas próximas N linhas, cada linha contém uma string e um número real, que indicam, respectivamente, a sigla da moeda estrangeira (com três letras maiúsculas) e a taxa de câmbio (100 unidades da moeda estrangeira em relação a HKD).

A última linha contém um número real, o valor de HKD que você pretende gastar em Hong Kong.

Saída

Seu programa deve produzir N linhas. Cada linha contém a sigla da moeda estrangeira e a quantidade desta moeda que você precisa ter para trocar pelo valor em HKD que você pretende gastar.

A ordem de saída das moedas estrangeiras deve ser a mesma da entrada.

Exemplos

Entrada	Saída
8 CNY 112.42 USD 776.22 GBP 973.50 AUD 589.95 CAD 587.70 EUR 826.20 CHF 774.05 BRR 111.10 1500.00	CNY 1334.28 USD 193.24 GBP 154.08 AUD 254.26 CAD 255.23 EUR 181.55 CHF 193.79 BRR 1350.14

Exercício 3)

Dada uma string contendo apenas os caracteres '(' e ')', determine se a string é válida. Os parênteses devem fechar na ordem correta, "()", "()()" e "()" são strings válidas; "(" e ")" não.

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 10$), o número de casos de teste.

Cada uma das N linhas seguintes contém uma string formada apenas pelos caracteres '(' e ')' de comprimento máximo igual a 10^4 .

Saída

Para cada caso de teste seu programa deve imprimir verdadeiro ou falso para indicar se a string é válida ou não.

Exemplos

Entrada	Saída
5 (() () (()) ()(verdadeiro verdadeiro verdadeiro falso falso

Exercício 4)

O jogo de Sudoku espalhou-se rapidamente por todo o mundo, tornando-se hoje o passatempo mais popular em todo o planeta. Muitas pessoas, entretanto, preenchem a matriz de forma incorreta, desrespeitando as restrições do jogo.

Sua tarefa neste problema é escrever um programa que verifica se uma matriz preenchida é ou não uma solução para o problema.

A matriz do jogo é uma matriz de inteiros 9 x 9. Para ser uma solução do problema, cada linha e coluna deve conter todos os números de 1 a 9. Além disso, se dividirmos a matriz em 9 regiões 3 x 3, cada uma destas regiões também deve conter os números de 1 a 9.

O exemplo abaixo mostra uma matriz que é uma solução do problema:

1	3	2	5	7	9	4	6	8
4	9	8	2	6	1	3	7	5
7	5	6	3	8	4	2	1	9
6	4	3	1	5	8	7	9	2
5	2	1	7	9	3	8	4	6
9	8	7	4	2	6	5	3	1
2	1	4	9	3	5	6	8	7
3	6	5	8	1	7	9	2	4
8	7	9	6	4	2	1	5	3

Entrada

A entrada contém 9 linhas, em que cada linha contém 9 números inteiros de 1 a 9 representando uma linha da matriz do Sudoku.

Saída

Seu programa deverá imprimir "SIM" se a matriz for a solução de um problema de Sudoku, e "NAO" caso contrário.

Exemplos

Entrada	Saída
1 3 2 5 7 9 4 6 8 4 9 8 2 6 1 3 7 5 7 5 6 3 8 4 2 1 9 6 4 3 1 5 8 7 9 2 5 2 1 7 9 3 8 4 6 9 8 7 4 2 6 5 3 1 2 1 4 9 3 5 6 8 7 3 6 5 8 1 7 9 2 4 8 7 9 6 4 2 1 5 3	SIM

Exercício 5)

Joãozinho está comprando os cadernos para o seu próximo curso: Análise de Algoritmos. Joãozinho sabe exatamente quantas páginas são necessárias para cada aula durante o semestre.

Tendo em vista a crise atual, Joãozinho só passa para um novo caderno quando todas as folhas do caderno atual tiverem sido preenchidas.

Joãozinho sempre compra cadernos com exatamente o mesmo número de páginas. Dadas quantas páginas são necessárias para cada uma das aulas do semestre, quantos cadernos Joãozinho precisa comprar para fazer as anotações de todas as aulas?

Entrada

A entrada contém duas linhas. A primeira linha da entrada contém dois inteiros N ($1 \leq N \leq 1000$), que é o número de aulas no semestre, e P ($1 \leq P \leq 1000$), que é o número de página de cada caderno. A segunda linha contém N inteiros A_1, A_2, \dots, A_N ($1 \leq A_i \leq P$, para $1 \leq i \leq N$), onde A_i é o número de páginas necessárias para a i -ésima aula.

Saída

Seu programa deve produzir uma linha contendo o número de cadernos que Joãozinho precisa para todo o semestre.

Exemplos

Entrada	Saída
4 100 60 30 20 60	2
3 100 70 100 35	3

Exercício 6)

Tri-du é um jogo de cartas derivado do popular jogo de Truco. O jogo utiliza um baralho normal de 52 cartas, com treze cartas de cada naipe, mas os naipes são ignorados. Apenas o valor das cartas, considerados como inteiros de 1 a 13, são utilizados.

No jogo, cada jogador recebe três cartas. As regras são simples:

- Um trio (três cartas de mesmo valor) ganha de uma dupla (duas cartas de mesmo valor).
- Um trio formado por cartas de maior valor ganha de um trio formado por cartas de menor valor.
- Uma dupla formada por cartas de maior valor ganha de uma dupla formada por cartas de menor valor.

Note que o jogo pode não ter ganhador em muitas situações; nesses casos, as cartas distribuídas são devolvidas ao baralho, que é embaralhado e uma nova partida é iniciada.

Ana e Beatriz são viciadas em Tri-du, mas como são ainda muito pequenas, às vezes têm dificuldade em determinar quem foi a vencedora de uma partida entre ambas. Sua tarefa é dada as cartas que Ana e Beatriz receberam uma partida de Tri-du, determine quem venceu, caso tenha havido um vencedor.

Entrada

A entrada consiste em duas linhas contendo três inteiros cada separados por um espaço em branco.

A primeira linha contém os inteiros A_1 , A_2 e A_3 ($1 \leq A_1, A_2, A_3 \leq 13$), representando as três cartas que Ana recebeu.

A segunda linha contém os inteiros B_1 , B_2 e B_3 ($1 \leq B_1, B_2, B_3 \leq 13$), representando as três cartas que Beatriz recebeu.

Saída

Seu programa deve produzir uma linha contendo uma única letra maiúscula: 'A' se Ana venceu a partida, 'B' se Beatriz venceu a partida ou 'E' se houve empate.

Exemplos

Entrada	Saída
10 7 10 5 5 5	B
7 7 7 4 4 4	A
13 10 1 4 5 9	E

Exercício 7)

Joãozinho está muito animado com a proximidade da Competição de Programação Cobol que acontecerá neste sábado. Ele quer encher o prédio da competição com frases e banners a respeito do evento. Por essa razão, Joãozinho contratou uma empresa para essa tarefa. Ele deu a empresa todas as frases que ele precisava e eles se responsabilizaram pelo processo de encher as paredes do prédio com as frases na forma de um grafiti.

O problema é que o pintor que foi enviado pela empresa tem uma desordem mental rara, Troca Incurável de Posições e Caracteres (TIPC). Por exemplo, ele pode trocar os caracteres de uma frase por outros ou pode trocar os caracteres da frase de posição. Após isso, as frases finais sequer guardam qualquer semelhança com as frases originais. Por exemplo, a frase “Bem_vindos_Competidores” pode se tornar “Estamos_otimos_aqui_:)”. Que coisa louca, não!

Mas pelo menos nós sabemos com certeza que a frase final terá sempre o mesmo número de caracteres da frase original. Agora Joãozinho que saber quantas posições têm caracteres diferentes a fim de corrigi-los.

Sua tarefa é determinar quantos caracteres precisam ser corrigidos antes que a Competição de Programação Cobol comece.

Entrada

A primeira linha da entrada contém um inteiro T ($1 \leq T \leq 100$) representando o número de frases. As próximas T linhas contêm duas strings cada separadas por um único espaço representando a frase original e a frase final, respectivamente.

As frases não são vazias e você pode assumir que elas são do mesmo tamanho e são compostas por no máximo 100 caracteres consecutivos sem espaços em branco.

Saída

Para cada par de frases da entrada seu programa deve produzir uma linha contendo o inteiro representando o número caracteres a ser corrigido antes que a Competição de Programação Cobol comece.

Exemplos

Entrada	Saída
4	8
Maratona Anotaram	22
Bemvindos_Competidores Estamos_otimos_aqui_:)	17
Welcome_Contestants We_are_fine_here_:)	23
Bienvenidos_Concursantes Estamos_muy_bien_aqui_:)	