

Centro de Educação Superior a Distância do
Estado do Rio de Janeiro – CEDERJ

Curso de Tecnologia em Sistemas de Computação – TSC

EAD-05.009 Fundamentos de Programação

Caderno de Exercícios

Aula 6

(Algoritmos de Busca, Busca do Menor e Maior Elementos)

Professores

Dante Corbucci Filho
Leandro A. F. Fernandes

Instruções

- Utilize Python 3 e a IDE PyCharm na elaboração de soluções para os problemas propostos;
- A entrada de cada problema deve ser lida da entrada padrão (teclado);
- A saída de cada problema deve ser escrita na saída padrão (tela);
- Siga o formato apresentado na descrição da saída, caso contrário não é garantido que a saída emitida será considerada correta;
- Na saída, toda linha deve terminar com o caractere `'\n'` ;
- Utilize o URI Online Judge (<http://www.urionlinejudge.com.br>) e submeta sua solução para correção automática.

Referências Autorais

Os exercícios apresentados nesta lista foram extraídos do URI Online Judge (<http://www.urionlinejudge.com.br>). Acesse a URL apresentada abaixo do título de cada problema para proceder com a correção automática de sua solução e, também, para consultar a autoria do enunciado.

Problema A: Frequência de Números

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1171>

Neste problema sua tarefa será ler vários números e em seguida dizer quantas vezes cada número aparece na entrada de dados, ou seja, deve-se escrever cada um dos valores distintos que aparecem na entrada por ordem crescente de valor.

Entrada

A entrada contém apenas 1 caso de teste. A primeira linha de entrada contém um único inteiro N , que indica a quantidade de valores que serão lidos para X ($1 \leq X \leq 2000$) logo em seguida. Com certeza cada número não aparecerá mais do que 20 vezes na entrada de dados.

Saída

Imprima a saída de acordo com o exemplo fornecido abaixo, indicando quantas vezes cada um deles aparece na entrada por ordem crescente de valor.

Exemplo

Entrada	Saída
7	4 aparece 1 vez(es)
8	8 aparece 2 vez(es)
10	10 aparece 3 vez(es)
8	260 aparece 1 vez(es)
260	
4	
10	
10	

Problema B: Letras

<https://www.urionlinejudge.com.br/judge/pt/problems/view/2457>

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. 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.

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.

Exemplo

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

Problema C: Loop Musical

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1089>

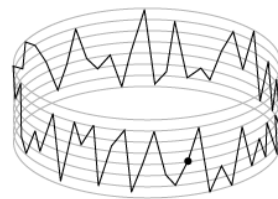
Um loop musical é um trecho de música que foi composto para repetir continuamente (ou seja, o trecho inicia novamente toda vez que chega ao final), sem que se note descontinuidade. Loops são muito usados na sonorização de jogos, especialmente jogos casuais pela internet.

Loops podem ser digitalizados por exemplo utilizando PCM. PCM, do inglês Pulse Code Modulation, é uma técnica para representação de sinais analógicos, muito utilizada em áudio digital. Nessa técnica, a magnitude do sinal é amostrada a intervalos regulares de tempo, e os valores amostrados são armazenados em sequência. Para reproduzir a forma de onda amostrada, o processo é invertido (demodulação).

Fernandinha trabalha para uma empresa que desenvolve jogos e compôs um bonito loop musical, codificando-o em PCM. Analisando a forma de onda do seu loop em um software de edição de áudio, Fernandinha ficou curiosa ao notar a quantidade de “picos” existentes. Um pico em uma forma de onda é um valor de uma amostra que representa um máximo ou mínimo local, ou seja, um ponto de inflexão da forma de onda. A figura abaixo ilustra (a) um exemplo de forma de onda e (b) o loop formado com essa forma de onda, contendo 48 picos.



(a) Uma forma de onda



(b) A mesma forma de onda como um loop

Fernandinha é uma amiga muito querida e pediu sua ajuda para determinar quantos picos existem no seu loop musical.

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um inteiro N , representando o número de amostras no loop musical de Fernandinha ($2 \leq N \leq 104$). A segunda linha contém N inteiros H_i , separados por espaços, representando a

sequência de magnitudes das amostras ($-104 \leq H_i \leq 104$ para $1 \leq i \leq N$, $H_1 \neq H_N$ e $H_i \neq H_{i+1}$ para $1 \leq i < N$). Note que H_1 segue H_N quando o loop é reproduzido.

O final da entrada é indicado por uma linha que contém apenas o número zero.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha, contendo apenas um inteiro, o número de picos existentes no loop musical de Fernandinha.

Exemplo

Entrada	Saída
2	2
1 -3	2
6	4
40 0 -41 0 41 42	
4	
300 450 449 450	
0	

Problema D: Diamantes e Areia

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1069>

João está trabalhando em uma mina, tentando retirar o máximo que consegue de diamantes " $\langle \rangle$ ". Ele deve excluir todas as partículas de areia "." do processo e a cada retirada de diamante, novos diamantes poderão se formar. Se ele tem como uma entrada $\langle \dots \langle \dots \rangle \dots \rangle \dots \rangle \dots$, três diamantes são formados. O primeiro é retirado de $\langle \dots \rangle$, resultando $\langle \dots \langle \dots \rangle \dots \rangle \dots \rangle \dots$. Em seguida o segundo diamante é retirado, restando $\langle \dots \dots \rangle \dots \rangle \dots$. O terceiro diamante é então retirado, restando no final $\dots \dots \rangle \dots$, sem possibilidade de extração de novo diamante.

Entrada

Deve ser lido um valor inteiro N que representa a quantidade de casos de teste. Cada linha a seguir é um caso de teste que contém até 1000 caracteres, incluindo "<,> ,."

Saída

Você deve imprimir a quantidade de diamantes possíveis de serem extraídos em cada caso de entrada.

Exemplo

Entrada	Saída
2 <...><...>> <<<...<.....<<<...>	3 1

Problema E: Detetive Watson

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1533>

John Watson, mesmo após anos trabalhando ao lado de Sherlock Holmes, nunca conseguiu entender como ele consegue descobrir quem é o assassino com tanta facilidade. Em uma certa noite, porém, Sherlock bebeu mais do que devia e acabou contando o segredo a John.

“Elementar, meu caro Watson”, disse Sherlock Holmes. “Nunca é o mais suspeito, mas sim o segundo mais suspeito”. Após descobrir o segredo, John decidiu resolver um crime por conta própria, só para testar se aquilo fazia sentido ou se era apenas conversa de bêbado.

Dada uma lista com N inteiros, representando o quanto cada pessoa é suspeita, ajude John Watson a decidir quem é o assassino, de acordo com o método citado.

Entrada

Haverá diversos casos de teste. Cada caso de teste inicia com um inteiro N ($2 \leq N \leq 1000$), representando o número de suspeitos.

Em seguida haverá N inteiros distintos, onde o i -ésimo inteiro, para todo $1 \leq i \leq N$, representa o quão suspeita a i -ésima pessoa é, de acordo com a classificação dada por John Watson. Seja V o valor do i -ésimo inteiro, $1 \leq V \leq 10000$.

O último caso de teste é indicado quando $N = 0$, o qual não deverá ser processado.

Saída

Para cada caso de teste imprima uma linha, contendo um inteiro, representando o índice do assassino, de acordo com o método citado.

Exemplo

Entrada	Saída
3	1
3 5 2	4
5	
1 15 3 5 2	
0	

Problema F: Procurando Subsequências

<https://www.urionlinejudge.com.br/judge/pt/problems/view/2126>

Dados dois números naturais N_1 e N_2 , diz-se que N_1 é subsequência contígua de N_2 se todos os dígitos de N_1 aparecem, na mesma ordem e de forma contígua, em N_2 . Crie uma aplicação que leia dois números naturais e diga se o primeiro é uma subsequência contígua do segundo.

Entrada

A entrada é composta por vários casos de teste e termina com final de arquivo (EOF). A primeira linha de cada entrada é composta por um valor natural N_1 ($1 < N_1 < 10^{10}$), a segunda linha é composta por um valor N_2 ($N_1 < N_2 < 10^{32}$).

Saída

Para cada caso de teste imprima a quantidade de subsequências contíguas e a posição onde a subsequência é iniciada, caso exista mais de uma subsequência, imprima onde é iniciada a última subsequência. Caso não exista subsequência, imprima "Nao existe subsequencia". Mostre o resultado conforme o exemplo de saída.

Exemplo

Entrada	Saída
78954 7895478954789547895447895478954 464133 1331646546874694 12 1231321455123214565423112	Caso #1: Qtd.Subsequencias: 6 Pos: 27 Caso #2: Nao existe subsequencia Caso #3: Qtd.Subsequencias: 3 Pos: 24