

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: Aumento de Salário

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1048>

Obs.: Resolva este problema utilizando busca em vetor com intervalos ao invés de encadear estruturas de seleção.

A empresa ABC resolveu conceder um aumento de salários a seus funcionários de acordo com a tabela abaixo:

| Salário | Percentual de Reajuste |
|-------------------|------------------------|
| 0.00 – 400.00 | 15% |
| 400.01 – 800.00 | 12% |
| 800.01 – 1200.00 | 10% |
| 1200.01 – 2000.00 | 7% |
| Acima de 2000.00 | 4% |

Leia o salário do funcionário e calcule e mostre o novo salário, bem como o valor de reajuste ganho e o índice reajustado, em percentual.

Entrada

A entrada contém apenas um valor de ponto flutuante, com duas casas decimais.

Saída

Imprima 3 linhas na saída: o novo salário, o valor ganho de reajuste e o percentual de reajuste ganho, conforme exemplo abaixo.

Exemplo

| Entrada | Saída |
|---------|--|
| 400.00 | Novo salario: 460.00 Reajuste ganho: 60.00 Em percentual: 15 % |

| Entrada | Saída |
|---------|--|
| 800.01 | Novo salario: 880.01 Reajuste ganho: 80.00 Em percentual: 10 % |

| Entrada | Saída |
|---------|---|
| 2000.00 | Novo salario: 2140.00 Reajuste ganho: 140.00 Em percentual: 7 % |

Problema B: DDD

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1050>

Obs.: Resolva este problema utilizando busca em vetor com intervalos ao invés de encadear estruturas de seleção.

Leia um número inteiro que representa um código de DDD para discagem interurbana. Em seguida, informe à qual cidade o DDD pertence, considerando a tabela abaixo:

| DDD | Destino |
|-----|----------------|
| 61 | Brasília |
| 71 | Salvador |
| 11 | Sao Paulo |
| 21 | Rio de Janeiro |
| 32 | Juiz de Fora |
| 19 | Campinas |
| 27 | Vitoria |
| 31 | Belo Horizonte |

Se a entrada for qualquer outro DDD que não esteja presente na tabela acima, o programa deverá informar: DDD nao cadastrado

Entrada

A entrada consiste de um único valor inteiro.

Saída

Imprima o nome da cidade correspondente ao DDD existente na entrada. Imprima DDD nao cadastrado caso não existir DDD correspondente ao número digitado.

Exemplo

| Entrada | Saída |
|---------|-----------|
| 11 | Sao Paulo |

Problema C: Substituição em Vetor I

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1172>

Faça um programa que leia um vetor $X[10]$. Substitua a seguir, todos os valores nulos e negativos do vetor X por 1. Em seguida mostre o vetor X .

Entrada

A entrada contém 10 valores inteiros, podendo ser positivos ou negativos.

Saída

Para cada posição do vetor, escreva “ $X[i] = x$ ”, onde i é a posição do vetor e x é o valor armazenado naquela posição.

Exemplo

| Entrada | Saída |
|---------|-------------|
| 0 | $X[0] = 1$ |
| -5 | $X[1] = 1$ |
| 63 | $X[2] = 63$ |
| 0 | $X[3] = 1$ |
| ... | ... |

Problema D: Menor e Posição

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1180>

Faça um programa que leia um valor N . Este N será o tamanho de um vetor $X[N]$. A seguir, leia cada um dos valores de X , encontre o menor elemento deste vetor e a sua posição dentro do vetor, mostrando esta informação.

Entrada

A primeira linha de entrada contém um único inteiro N ($1 < N < 1000$), indicando o número de elementos que deverão ser lidos em seguida para o vetor $X[N]$ de inteiros. A segunda linha contém cada um dos N valores, separados por um espaço.

Saída

A primeira linha apresenta a mensagem “Menor valor:” seguida de um espaço e do menor valor lido na entrada. A segunda linha apresenta a mensagem “Posicao:” seguido de um espaço e da posição do vetor na qual se encontra o menor valor lido, lembrando que o vetor inicia na posição zero.

Exemplo

| Entrada | Saída |
|-----------------------------|-------------------------------|
| 10 1 2 3 4 -5 6 7 8 9 10 | Menor valor: -5 Posicao: 4 |

Problema E: 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 |

Problema F: 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

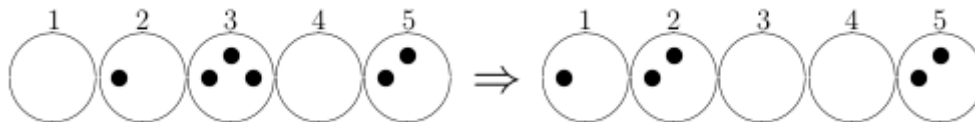
| Entrada | Saída |
|------------|-------|
| 3 | 1 |
| 3 5 2 | 4 |
| 5 | |
| 1 15 3 5 2 | |
| 0 | |

Problema G: Awari 2.0

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1998>

Awari é um jogo de um jogador das Antilhas, que é jogado com caixas e pedras em vez de cartões. Outra versão de Awari é jogado com N caixas numeradas de 1 a N , cada uma contendo, no início do jogo, zero ou mais pedras. As regras deste jogo são muito simples, porque há apenas um tipo de movimento válido, que consiste em escolher uma caixa numerada i que contém exatamente i pedras, e em seguida, pegar essas pedras da caixa, a fim de usá-las para adicionar uma única pedra para cada caixa numerada de 1 a $i - 1$; a pedra restante é mantida pelo jogador. Estes movimentos são aplicados em sucessão, desde que exista uma caixa i que contém exatamente i pedras. Quando isso não é mais verdade, o jogo termina. O jogador ganha se, nesta fase, cada caixa está vazia, e perde caso contrário.

Na figura a seguir, no lado esquerdo há um possível estado inicial de um jogo com $n = 5$ caixas (os círculos) contendo $P_1 = 0$, $P_2 = 1$, $P_3 = 3$, $P_4 = 0$ e $P_5 = 2$ pedras (os pontos pretos). Se a caixa de número 3, que contém $P_3 = 3$ pedras, foi escolhida para dar o próximo passo, a configuração resultante seria a do mostrado no lado direito da figura. Além disso, para o jogador teria agora uma pedra em seu poder.



Dado o estado inicial das caixas, você deve determinar se é possível ganhar o jogo, ou seja, se houver uma sequência de movimentos válidos que depois de todas as caixas são deixadas vazias.

Entrada

Cada caso de teste é descrito usando duas linhas. A primeira linha contém um inteiro N , indicando o número de caixas ($1 \leq N \leq 500$). A segunda linha contém N números inteiros P_i , que representa o número de pedras nas caixas no início do jogo, a partir da caixa 1 para a caixa N , respectivamente ($0 \leq P_i \leq 500$ para $i = 1, \dots, N$). Em cada caso de teste que haja pelo menos uma caixa não vazia, isto é, existe i de 1 a N tal que $P_i \neq 0$. O fim da linha de entrada é uma linha que contém o número -1 .

Saída

Para cada caso de teste, você deve imprimir uma única linha contendo um único caractere. Este caractere deve ser a letra maiúscula 'S' se é possível ganhar o jogo; Caso contrário, deve-se a letra maiúscula 'N'.

Exemplo

| Entrada | Saída |
|-----------|-------|
| 5 | N |
| 0 1 3 0 2 | S |
| 4 | N |
| 1 1 3 0 | |
| 3 | |
| 1 2 3 | |
| -1 | |

Problema H: Subsequências

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1507>

Dado duas sequências, diga se a segunda é uma subsequência da primeira.

Entrada

A entrada começa com um inteiro N ($0 < N \leq 10$), o número de casos de teste. Seguem N casos testes. Um caso de teste começa com uma sequência de caracteres S (S em $[a - z A - Z]$, $0 < |S| < 100000$). Segue um inteiro Q ($0 < Q \leq 1000$), o número de queries. As próximas Q linhas são compostas por uma sequência de caracteres R (R em $[a - z A - Z]$, $0 < |R| < 100$) cada.

Saída

Para cada query R , imprima “Yes” (sem aspas) se R for uma subsequência de S , ou “No” (sem aspas), caso contrário.

Exemplo

| Entrada | Saída |
|----------|-------|
| 1 | Yes |
| aabccbba | No |
| 2 | |
| abc | |
| abbc | |

Problema I: Parafusos e Porcas

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1520>

Pedro está trabalhando em um novo emprego em uma loja de parafusos e porcas. Toda semana ele irá receber novas encomendas de parafusos mistos em grandes caixas e precisa saber onde ele deve armazenar esses produtos.

Mango, que é supervisor de Pedro, pediu-lhe para organizar todos os parafusos e porcas em um rack enorme, de forma que seja fácil de responder onde é possível encontrar cada predeterminado lote de parafusos e quanto de cada um destes lotes a loja tem.

Pedro recebe todos os parafusos e porcas em caixas etiquetadas com o intervalo de tamanho dos produtos. Por exemplo, duas caixas com os respectivos intervalos (1,2) e (4,8) representam lotes de produtos com tamanho {1,2,4,5,6,7,8}. Um número de parafuso e porca pode estar presente em mais de uma caixa, e ele aparece na lista uma vez para cada intervalo. Por exemplo, três caixas com os respectivos intervalos (1,3), (2,4) e (3,5) representam lotes de parafusos e porcas com tamanho {1,2,2,3,3,3,4,4,5}.

Entrada

A entrada contém vários casos de teste e termina com EOF. Cada caso de teste é composto por várias linhas. A primeira linha contém um inteiro positivo N ($N < 100$) que indica a quantidade de caixas de parafusos. Seguem N linhas, cada uma delas com dois valores X e Y ($1 \leq X < Y \leq 100$) que representam os tamanhos dos lotes de parafusos e porcas presentes naquela caixa. A próxima linha de entrada conterá um único inteiro positivo Num ($1 \leq Num \leq 100$), que indica o número que Mango deseja pesquisar após todos os produtos serem organizados na estante ou rack.

Obs.: A estante ou rack possui prateleiras enumeradas de 0 a P , sendo que este P não deve ser superior a 10000.

Saída

Cada caso de entrada deverá produzir uma única linha de saída, indicando as posições da prateleira nas quais o parafuso de tamanho Num se encontra ou indicando que não foi possível encontrar o parafuso, conforme o exemplo abaixo.

Exemplo

| Entrada | Saída |
|---------|---------------------|
| 3 | 4 found from 6 to 7 |
| 1 3 | 3 found from 5 to 8 |
| 2 4 | 7 not found |
| 3 5 | |
| 4 | |
| 4 | |
| 1 3 | |
| 2 4 | |
| 1 5 | |
| 3 6 | |
| 3 | |
| 2 | |
| 1 3 | |
| 3 5 | |
| 7 | |

Problema J: 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 | |