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 09

(Persistência de Dados: Arquivo Texto)

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: Encaixa ou Não II

https://www.urionlinejudge.com.br/judge/pt/problems/view/1241

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado entrada.txt e a sua saída deve ser escrita no arquivo saída.txt.

Paulinho tem em suas mãos um novo problema. Agora a sua professora lhe pediu que construísse um programa para verificar, à partir de dois valores muito grandes A e B, se B corresponde aos últimos dígitos de A.

Entrada

A entrada consiste de 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 consiste de dois valores **A** e **B** maiores que zero, cada um deles podendo ter até 1000 dígitos.

Saída

Para cada caso de entrada imprima uma mensagem indicando se o segundo valor encaixa no primeiro valor, conforme exemplo abaixo.

Exemplo

Entrada	Saída
4 562345234857238547555454545478690 78690 5434554 543 1243 1243 54 64545454545454545454545454545454554	encaixa nao encaixa encaixa nao encaixa

Problema B: Instruções do Robô

https://www.urionlinejudge.com.br/judge/pt/problems/view/1574

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado entrada.txt e a sua saída deve ser escrita no arquivo saída.txt.

Você possui um robô na origem do eixo *x*. O robô receberá algumas instruções. Sua tarefa é predizer sua posição depois de executar todas as instruções.

- LEFT: move uma unidade para a esquerda (diminui *p* em 1, onde *p* é a posição do robô antes de mover)
- RIGHT: move uma unidade para a direita (incrementa *p* em 1)
- SAME AS *i*: executa a mesma ação que na *i*-ésima instrução. É garantido que *i* é um inteiro positivo não maior que o número de instruções já executadas.

Entrada

A primeira linha contém o número de casos de testes *T (T <= 100)*. Cada caso de teste inicia com um inteiro *n (1 <= n <= 100)*, o número de instruções. Cada uma das *n* linhas seguintes contém uma instrução.

Saída

Para cada caso de teste, imprima a posição final do robô. Note que após processar cada caso de teste, o robô deve ter sua posição inicial resetada para a origem.

Exemplo

Entrada	Saída
2 3 LEFT RIGHT SAME AS 2 5 LEFT SAME AS 1 SAME AS 2 SAME AS 1 SAME AS 4	1 -5

Problema C: Bits Trocados

https://www.urionlinejudge.com.br/judge/pt/problems/view/2187

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado entrada.txt e a sua saída deve ser escrita no arquivo saída.txt.

As Ilhas Weblands formam um reino independente nos mares do Pacífico. Como é um reino recente, a sociedade é muito influenciada pela informática. A moeda oficial é o Bit; existem notas de B$ 50,00, B$10,00, B$5,00 e B$1,00. Você foi contratado(a) para ajudar na programação dos caixas automáticos de um grande banco das Ilhas Weblands.

Os caixas eletrônicos das Ilhas Weblands operam com todos os tipos de notas disponíveis, mantendo um estoque de cédulas para cada valor (B$ 50,00, B$10,00, B$5,00 e B$1,00). Os clientes do banco utilizam os caixas eletrônicos para efetuar retiradas de um certo número inteiro de Bits.

Sua tarefa é escrever um programa que, dado o valor de Bits desejado pelo cliente, determine o número de cada uma das notas necessário para totalizar esse valor, de modo a minimizar a quantidade de cédulas entregues. Por exemplo, se o cliente deseja retirar B$50,00, basta entregar uma única nota de cinquenta Bits. Se o cliente deseja retirar B$72,00, é necessário entregar uma nota de B$50,00, duas de B$10,00 e duas de B$1,00.

Entrada

A entrada é composta de vários conjuntos de teste. Cada conjunto de teste é composto por uma única linha, que contém um número inteiro positivo **V** ($0 \leq \mathbf{V} \leq 10000$), que indica o valor solicitado pelo cliente. O final da entrada é indicado por **V** = 0.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste **n**”, onde **n** é numerado a partir de 1. Na segunda linha devem aparecer quatro inteiros **I**, **J**, **K** e **L** que representam o resultado encontrado pelo seu programa: **I**indica o número de cédulas de B$50,00, **J** indica o número de cédulas de B$10,00, **K** indica o número de cédulas de B$5,00 e **L** indica o número de cédulas de B$1,00. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

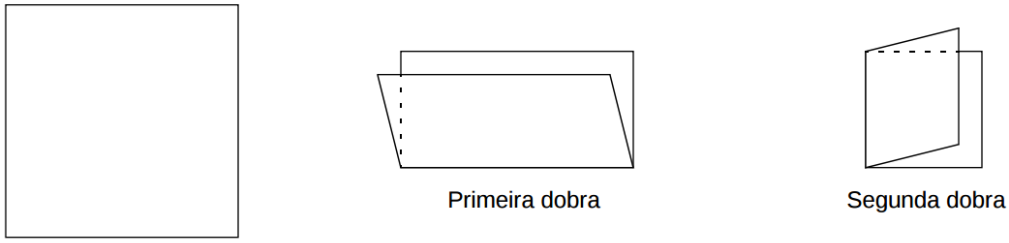
Entrada	Saída
1 72 0	Teste 1 0 0 0 1 Teste 2 1 2 0 2

Problema D: Dobradura

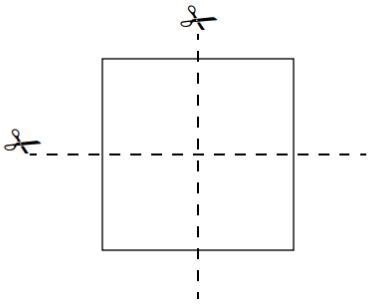
https://www.urionlinejudge.com.br/judge/pt/problems/view/2229

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado entrada.txt e a sua saída deve ser escrita no arquivo saída.txt.

Zezinho tem aulas de Iniciação Artística em sua escola, e recentemente aprendeu a fazer dobraduras em papel. Ele ficou fascinado com as inúmeras possibilidades de se dobrar uma simples folha de papel. Como Zezinho gosta muito de matemática, resolveu inventar um quebra-cabeça envolvendo dobraduras. Zezinho definiu uma operação de dobradura D que consiste em dobrar duas vezes uma folha de papel quadrada de forma a conseguir um quadrado com 1/4 do tamanho original, conforme ilustrado na figura.



Depois de repetir N vezes esta operação de dobradura D sobre o papel, Zezinho cortou o quadrado resultante com um corte vertical e um corte horizontal, conforme a figura abaixo.



Zezinho lançou então um desafio aos seus colegas: quem adivinha quantos pedaços de papel foram produzidos?

Entrada

A entrada é composta de vários conjuntos de teste. Cada conjunto de teste é composto de uma única linha, contendo um número inteiro **N** (-1 ≤ **N** ≤ 15) que indica o número de vezes que a operação de dobradura **D** foi aplicada. O final da entrada é indicado por **N** = -1.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde

n é numerado a partir de 1. A segunda linha deve conter o número de pedaços de papel obtidos depois de cortar a dobradura, calculado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada	Saída
1 0 -1	Teste 1 9 Teste 2 4

Problema E: Domingo de Manhã

https://www.urionlinejudge.com.br/judge/pt/problems/view/2242

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado entrada.txt e a sua saída deve ser escrita no arquivo saída.txt.

Em chats, é muito comum entre jovens e adolescentes utilizar sequências de letras, que parecem muitas vezes aleatórias, para representar risadas. Alguns exemplos comuns são:

huaauhahhuahau
hehehehe
ahahahaha
jaisjjkasjksjjskjakijs
huehuehue

Cláudia é uma jovem programadora que ficou intrigada pela sonoridade das "risadas digitais". Algumas delas ela nem mesmo consegue pronunciar! Mas ela percebeu que algumas delas parecem transmitir melhor o sentimento da risada que outras. A primeira coisa que ela percebeu é que as consoantes não interferem no quanto as risadas digitais influenciam na transmissão do sentimento. A segunda coisa que ela percebeu é que as risadas digitais mais engra¸cadas são aquelas em que as sequências de vogais são iguais quando lidas na ordem natural (da esquerda para a direita) ou na ordem inversa (da direita para a esquerda), ignorando as consoantes. Por exemplo, "hahaha" e "huaauhahhuahau" estão entre as risadas mais engraçadas, enquanto "riajkjdhhihhjak" e "huehuehue" não estão entre as mais engraçadas.

Cláudia está muito atarefada com a análise estatística das risadas digitais e pediu sua ajuda para escrever um programa que determine, para uma risada digital, se ela é das mais engraçadas ou não.

Entrada

A entrada é composta por uma linha, contendo uma sequência de no máximo 50 caracteres, formada apenas por letras minúsculas sem acentuação. As vogais são as letras 'a','e','i','o','u'. A sequência contém pelo menos uma vogal.

Saída

Seu programa deve produzir uma linha contendo um caractere, "S" caso a risada seja das mais engra¸cadas, ou "N" caso contrário.

Exemplo

Entrada	Saída
hahaha	S

Entrada	Saída
riajkjdhhihhjak	N

Entrada	Saída
a	S

Entrada	Saída
huaauhahhuahau	S

Problema F: Cofrinhos da Vó Vitória

https://www.urionlinejudge.com.br/judge/pt/problems/view/2247

Observação: Após o seu programa funcionar no URI, carregue vários valores de entrada através de um arquivo chamado entrada.txt e a sua saída deve ser escrita no arquivo saída.txt.

Vó Vitória mantém, desde o nascimento dos netos Joãozinho e Zezinho, um ritual que faz a alegria dos meninos. Ela guarda todas as moedas recebidas como troco em dois pequenos cofrinhos, um para cada neto. Quando um dos cofrinhos fica cheio, ela chama os dois netos para um alegre almoço, ao final do qual entrega aos garotos as moedas guardadas nos cofrinhos de cada um.

Ela sempre foi muito zelosa quanto à distribuição igualitária do troco arrecadado. Quando, por força do valor das moedas, ela não consegue depositar a mesma quantia nos dois cofrinhos, ela memoriza a diferença de forma a compensá-la no próximo depósito.

Vó Vitória está ficando velha e tem medo que deslizes de memória a façam cometer injustiças com os netos, deixando de compensar as diferenças entre os cofrinhos. Sua tarefa é ajudar Vó Vitória, escrevendo um programa de computador que indique as diferenças entre os depósitos, de forma que ela não tenha que preocupar-se em memorizá-las.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém um número inteiro **N** ($0 \leq \mathbf{N} \leq 100$), que indica o número de depósitos nos cofrinhos. As **N** linhas seguintes descrevem cada uma um depósito nos cofrinhos; o depósito é indicado por dois valores inteiros **J** e **Z** ($0 \leq \mathbf{J},\mathbf{Z} \leq 100$), separados por um espaço em branco, representando respectivamente os valores, em centavos, depositados nos cofres de Joãozinho e Zezinho. O final da entrada é indicado por **N** = 0.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir um conjunto de linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado seqüencialmente a partir de 1. A seguir seu programa deve escrever uma linha para cada depósito do conjunto de testes. Cada linha deve conter um inteiro que representa a diferença (em centavos) entre o valor depositado nos cofrinhos do Joãozinho e do Zezinho. Deixe uma linha em branco ao final de cada conjunto de teste. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada	Saída
3 20 25 10 5 10 10 4 0 5 12 0 0 20 17 1 0	Teste 1 -5 0 0 Teste 2 -5 7 -13 3