27/05/2019

**CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - UNIFEOB**

**AVALIAÇÃO LTP I**

**PROF. MAX STREICHER VALLIM**

**Exercício 1: Contando as Letras**

Um pesquisador linguístico está escrevendo um artigo sobre como as diferentes línguas baseadas no alfabeto romano tem sido utilizada desde sua origem. Em uma das seções do artigo o pesquisador queria estabelecer as diferença entre essas línguas. O primeiro índice a ser avaliado era a diferença da intensidade da utilização de vogais e consoantes entre elas. O pesquisador então coletou vários textos escritos em diferentes línguas baseadas nesse alfabeto. Selecionou aleatoriamente vários trechos desses textos e para fazer a contagem das vogais e consoantes contatou o departamento de informática em busca de algum aluno que pudesse desenvolver um programa capaz de realizar essa contagem. Rapidamente nosso departamento respondeu e indicou você para ajudar o pesquisador. Contribua com o trabalho de pesquisa e orgulhe o nosso departamento.

**Entrada**

A entrada começa com um número 1 <= N <= 1000 , que representa o número de casos de teste. Então seguem N linhas, cada uma com um trecho de máximo 80 caracteres. Cada trecho só pode ser formado por caracteres alfanuméricos, caracteres de pontuação ('?', '!', '.' e ',' apenas) e espaços em branco. As letras são sempre minúsculas e não há letras acentuadas. Leia da entrada padrão.

**Saída**

Para cada caso de teste da entrada, imprima uma linha na saída contendo, respectivamente, a quantidade de vogais e consoantes. Use a saída padrão.

Exemplo de entrada respectiva saída

ola mundo!   4 4

hello world! 3 7

Autor do Problema: Eduardo Araújo, finalista da Maratona de Programação da SBC 2007.

**Exercício 2: Problema dos Reis**

Um problema bem conhecido entre os estudantes e professores da área de computação é o problema das 8 rainhas. Os primeiros relatos deste problema datam do início do século, estudado por Gauss e bem conhecido dos livros que tratam de combinatória ou matemática recreativa.

O problema consiste em dispor oito 8 rainhas em um tabuleiro 8x8 de tal modo que elas não se ataquem.

Baseado neste problema, Joãozinho resolveu inventar um novo problema: O Problema dos Reis. Neste novo problema, Joãozinho está interessado em saber quantos reis, no máximo, é possível colocar em um tabuleiro de forma que nenhum rei ataque o outro.

Obs: Um rei ataca somente as casas vizinhas à qual ele se encontra.

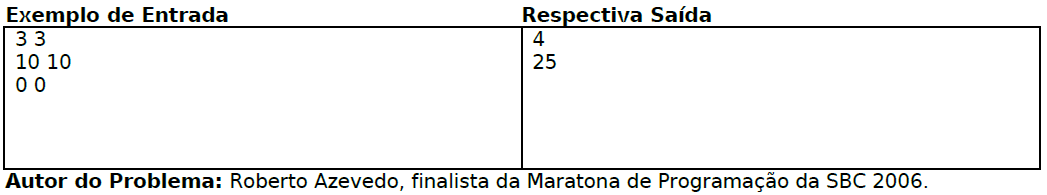
**Entrada**

A entrada é composta por vários casos de teste. Cada caso de teste inicia-se com uma linha contendo dois inteiros **L** e **C** que representam as dimensões (largura e comprimento) do tabuleiro. A entrada termina quando **L = 0** e **C = 0,** a qual não deve ter uma saída correspondente. **Leia da entrada padrão.**

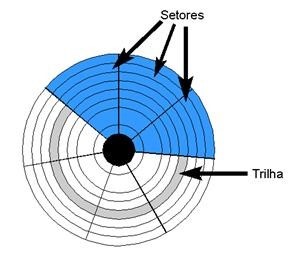
Limites: 0 <= **L** <= 10000, 0 <= **C** <= 10000.

**Saída**

Para cada caso de teste da entrada imprima uma linha na saída que contenha o número máximo de reis que podem ser colocados neste tabuleiro de forma que nenhum ataque um outro. **Use a saída padrão.**

****

**Exercício 3: Setores no Disco**



Vocês se lembram como aquele velho LP que vocês tinham em casa funcionava? Ele usava um cabeçote posicionado em um certo lugar no disco para ler uma informação. É incrível como muitos dispositivos de armazenamento (Discos Rígidos, CDs, DVDs) ainda usam a mesma ideia.

Entretanto, há diferenças na forma de armazenamento da informação. No seu HD, por exemplo, cada disco é dividido em trilhas, como se fosse cada circunferência concêntrica ao disco. E cada trilha é dividida em setores, que são uma fração dessa trilha. O engraçado é que os setores de uma trilha não são numerados na ordem em que eles estão dispostos. O motivo disso é porque em cada setor é lido de uma só vez, mas entre acabar uma leitura e começar outra existe um intervalo. Enquanto esse tempo passa, o disco continua girando e passa por alguns setores até voltar a ler de novo. Por exemplo, um disco tem um atraso de um setor pra voltar a ler a trilha e esta tem 7 setores. Se os setores estivessem organizados em ordem, e queremos ler o setor 1 e 2, leríamos o 1, o disco não leria o 2 porque está no intervalo e só voltaria a ler quando estivesse no setor 3. Nesse exemplo, ele teria que esperar uma volta no disco, mas se o disco fosse organizado da forma 1, 5, 2, 6, 3, 7, 4, quando o disco voltasse a ler, ele estaria exatamente no setor 2.Você foi contratado por uma empresa para fazer uma rotina no seu Sistema Operacional que numera os setores automaticamente.

# Entrada

No começo do programa será informado um inteiro que é a quantidade casos de teste. Para cada caso serão informados 2 números inteiros representando respectivamente o tamanho da trilha e o atraso(em número de setores) de voltar a ler, em que ambos são estritamente positivos menores que 50. **Leia da entrada padrão.**

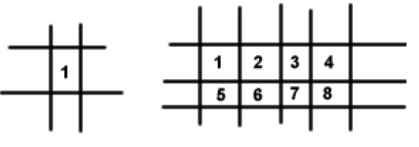
# Saída

A saída deve ser a melhor forma de se organizar o disco começando do setor 1. Veja o exemplo de saída. **Use a saída padrão.**



**Exercício 4: Retângulos**

Como trabalho final da disciplina de Introdução à Informática, o professor resolveu passar um trabalho que envolvesse toda a turma. Ele começou a escrever várias perguntas estranhas no quadro e assim que terminou disse que o trabalho final era fazer uma biblioteca de algoritmos que respondessem todas aquelas perguntas. A verdadeira lição não era as questões em si, mas o trabalho em equipe e como a expressão “dividir para conquistar” deve fazer parte da vida de um analista de sistemas. Apesar de claramente exigir um pouco de raciocínio, os alunos logo começaram a discutir sobre a utilidade de uma biblioteca para aquelas perguntas estranhas. Ignorando os comentários, o professor divide as questões entre os alunos e avisa que a biblioteca tem que estar completa para que todos possam ser aprovados.

José acabou pegando uma bem fácil: “Se traçarmos X linhas horizontais e Y linhas verticais, quantos retângulos formaremos?”. Antes mesmo de ir para casa ele pegou uma folha de papel e começou a fazer alguns desenhos. Através deles, visualizar a quantidade de retângulo ficou muito fácil e logo conseguiu generalizar o caso para que pudesse desenvolver o algoritmo. Como não podia falhar com seus amigos, ele tinha que ter certeza que seu algoritmos está funcionando. Então, preparou uma bateria de teste e pediu que você resolvesse o mesmo problema para que pudesse comparar seus resultados.

Ajude José e consequentemente sua turma a passar na disciplina.

Considere que as linhas nunca são coincidentes e que retângulos formados pela união de 2 ou mais retângulos não devem ser contados.

**Entrada**

A entrada começa com um número 1 <= N <= 100,, que representa a quantidade de casos de teste. Então seguem N linhas, cada uma contendo o número 0 <= H <= 100 de linhas horizontais e um número 0 <= V<= 100 de linhas verticais. **Leia da entrada padrão.**

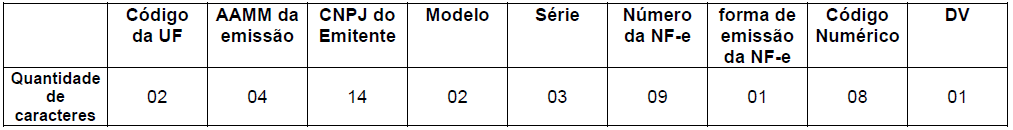
**Saída**

Para cada caso de teste da entrada, imprima uma linha na saída contendo a quantidade retângulos que foram formados pelas linhas. **Use a saída padrão.**



**Exercício 5: Dígito verificador da chave de uma NFe (Nota Fiscal Eletrônica)**

A NFe foi desenvolvida para substituir as tradicionais Notas Fiscais em papel. Atualmente, grande parte das empresas é obrigada a emitir esses documentos e a tendência é que em alguns anos todas as empresas passem a utilizar essa sistemática. A NFe é um arquivo XML com informações das operações comerciais realizadas pela empresa, como por exemplo, vendas. O XML contém todas as informações da operação comercial. Dentre as informações, podem-se destacar os dados do emitente, do destinatário, dos produtos e da tributação. Um XML de uma NFe é único, sendo identificado por uma chave de 44 dígitos, que é composta da seguinte forma:



35 1501 02472357000106 55 001 000016399 1 00411903 **6**

(chave da NFe 000016399)

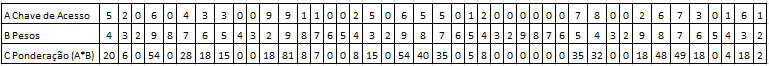
Seu desafio é elaborar um programa para calcular o dígito verificador da chave de uma NFe.

O dígito verificador da chave da NFe é baseado em um cálculo do módulo 11. O módulo 11 de um número é calculado multiplicando-se cada algarismo pela sequência de multiplicadores 2,3,4,5,6,7,8,9,2,3, ... posicionados da direita para a esquerda. A somatória dos resultados das ponderações dos algarismos é dividida por 11 e o DV (dígito verificador) será a diferença entre o divisor (11) e o resto da divisão:

DV = 11 - (resto da divisão).

Quando o resto da divisão for 0 (zero) ou 1 (um), o DV deverá ser igual a 0 (zero).

Exemplo: consideremos que a chave de acesso tem a seguinte sequência de caracteres:



Somatória das ponderações = 644

Dividindo a somatória das ponderações por 11 teremos, 644 /11 = 58 restando 6.

Como o dígito verificador DV = 11 - (resto da divisão), portando 11 - 6 = 5 .

Neste caso o DV da chave de acesso da NF-e é igual a "5", valor este que deverá compor a chave de acesso, totalizando a uma sequência de 44 caracteres.

**Entrada**

Deve ser a chave com 43 dígitos. **Leia da Entrada Padrão.**

**Saída**

Deve ser a chave com 44 dígitos, ou seja, incluindo o DV. **Use a saída padrão.**

**Autor do Problema:** Max Streicher Vallim, professor do curso de ADS-UNIFEOB.