



OBI2014

Caderno de Tarefas

Modalidade Programação • Nível 2, Fase 1

10 de maio de 2014

A PROVA TEM DURAÇÃO DE 5 HORAS

Promoção:



Sociedade Brasileira de Computação

Patrocínio:



Fundação Carlos Chagas

Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 9 páginas (não contando a folha de rosto), numeradas de 1 a 9. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Pascal devem ser arquivos com sufixo *.pas*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; e soluções na linguagem Python devem ser arquivos com sufixo *.py*. Para problemas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada problema.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou disquete, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
 - em Pascal: *readln*, *read*, *writeln*, *write*;
 - em C: *scanf*, *getchar*, *printf*, *putchar*;
 - em C++: as mesmas de C ou os objetos *cout* e *cin*.
 - em Java: qualquer classe ou função padrão, como por exemplo *Scanner*, *BufferedReader*, *BufferedWriter* e *System.out.println*
 - em Python: *read*, *readline*, *readlines*, *print*, *write*
- Procure resolver o problema de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

Copa do Mundo

Nome do arquivo fonte: `copa.c`, `copa.cpp`, `copa.pas`, `copa.java`, ou `copa.py`

A Nlogônia é atualmente um dos países com maior crescimento econômico no mundo, e seus governantes têm se esforçado para que o país seja mais conhecido e respeitado internacionalmente. Recentemente a Nlogônia foi escolhida para ser a sede da Copa do Mundo de Futebol Amador, e está se preparando para receber os milhares de torcedores que o evento atrai.

Como parte da preparação para a Copa, o governo planeja realizar uma reforma em todo o sistema de transporte intermunicipal, que é hoje composto de uma malha de rodovias e ferrovias, cada rodovia ou ferrovia interligando um par de cidades. Com as rodovias e ferrovias existentes já é possível viajar entre qualquer par de cidades (possivelmente passando por outras cidades no caminho), mas o governo quer oferecer melhores condições de transporte para os visitantes e a população.

Como não há recursos para reformar todas as rodovias e ferrovias, o governo quer escolher um conjunto de rodovias e ferrovias para ser reformado, e já realizou um estudo para estabelecer o custo de reforma de cada rodovia e ferrovia. A escolha deve obedecer aos seguintes critérios:

1. ao final da reforma, deve ser possível viajar entre qualquer par de cidades (possivelmente passando por outras cidades) utilizando apenas rodovias ou ferrovias reformadas;
2. para priorizar o transporte público, dentre as escolhas que satisfazem a restrição 1, deve-se escolher uma que minimize o número de rodovias reformadas;
3. dentre as escolhas que satisfazem as restrições 1 e 2, deve-se escolher uma que minimize o custo total.

Você foi contratado para escrever um programa que, conhecidos os custos de reforma de cada rodovia e ferrovia, determine o menor custo possível para a reforma, obedecendo os critérios estabelecidos.

Entrada

A primeira linha da entrada contém três inteiros N , F e R , indicando respectivamente o número de cidades, de ferrovias e de rodovias. As cidades são identificadas por inteiros de 1 a N . Cada uma das F linhas seguintes descreve uma ferrovia e contém três inteiros A , B e C , onde A e B representam cidades e C representa o custo da reforma da ferrovia que interliga A e B . Cada uma das R linhas seguintes descreve uma rodovia e contém três inteiros I , J e K , onde I e J representam cidades e K representa o custo da reforma da rodovia que interliga I e J .

Saída

Seu programa deve produzir uma única linha, contendo o menor custo possível para o conjunto de reformas de ferrovias e rodovias, obedecendo aos critérios estabelecidos.

Restrições

- $2 \leq N \leq 100$; $1 \leq F \leq N(N - 1)/2$; $1 \leq R \leq N(N - 1)/2$
- $1 \leq A < B \leq N$ e $1 \leq I < J \leq N$
- $1 \leq C \leq 1000$ e $1 \leq K \leq 1000$

Informações sobre a pontuação

- Para um conjunto de casos de testes totalizando 20 pontos, $2 \leq N \leq 6$.

Exemplos

Entrada	Saída
3 3 2 1 2 1000 1 3 1000 2 3 900 1 3 800 2 3 700	1900

Entrada	Saída
5 4 5 3 4 300 1 2 100 2 4 300 1 3 250 4 5 600 3 4 200 2 3 100 2 5 400 1 5 450	1050

Entrada	Saída
5 2 3 4 5 60 2 3 60 1 2 50 1 4 50 3 4 50	220

Língua do P

Nome do arquivo fonte: `lingua.c`, `lingua.cpp`, `lingua.pas`, `lingua.java`, ou `lingua.py`

Uma brincadeira que crianças adoram é se comunicar na *língua do P*, acrescentando *pê* antes de cada sílaba, como uma forma de código para dificultar que outras pessoas entendam a conversa (pê-va pê-mos pê-no pê-ci pê-ne pê-ma?).

Jacy e Kátia adaptaram a língua do P para mensagens eletrônicas, acrescentando a letra P minúscula ‘p’ antes de cada letra das palavras de uma mensagem. Um exemplo de mensagem codificada e a respectiva mensagem decodificada é mostrada na figura abaixo.

Mensagem codificada	Mensagem decodificada
pVpappops papo pcipnpepmpa	Vamos ao cinema

Sua tarefa é escrever um programa que decodifique uma mensagem escrita na língua do P eletrônica de Jacy e Kátia.

Entrada

A entrada consiste de uma única linha, contendo uma mensagem escrita na língua do P eletrônica de Jacy e Kátia.

Saída

Seu programa deve produzir uma única linha, contendo a mensagem decodificada.

Restrições

- A mensagem contém apenas letras maiúsculas e minúsculas e espaços em branco.
- A mensagem tem entre 1 e 1000 caracteres.
- Não há dois espaços em branco consecutivos na mensagem.

Exemplos

Entrada	Saída
pÜpm pfpiplpmpe plpepgpapl	Um filme legal

Entrada	Saída
pA pppapppa pdpo pPpapppa	A papa do Papa

PacMan

Nome do arquivo fonte: `pacman.c`, `pacman.cpp`, `pacman.pas`, `pacman.java`, ou `pacman.py`

Pacman é um jogo muito conhecido, onde o personagem tenta comer a maior quantidade possível de bolinhas, tendo ao mesmo tempo que fugir de vários fantasmas. Dessa vez, nosso personagem quer carregar a comida coletada para casa, mas o encontro com um fantasma, ao invés de terminar o jogo, faz com que toda a comida coletada seja roubada.

Neste problema os fantasmas não se movem, e o jogador sempre faz o Pacman percorrer o seguinte caminho:

1. O Pacman começa no canto superior esquerdo do tabuleiro.
2. O Pacman percorre toda a linha, da esquerda para direita, até chegar ao lado direito do tabuleiro.
3. O jogador desce uma posição, e percorre toda a linha, desta vez da direita para a esquerda.
4. As etapas 2 e 3 se repetem até que todo o tabuleiro tenha sido percorrido.

Infelizmente, Pacman não pode ignorar os comandos do usuário para fugir dos fantasmas ou pegar mais comida, mas ele pode, a qualquer momento, se aproveitar de um bug de implementação e interromper o jogo, levando consigo toda a comida que estiver carregando.

Você deve escrever um programa que determine a maior quantidade de comida que o Pacman pode levar, se escolher a melhor hora possível para sair. Note que o jogador também tem a opção de não sair antes do final do jogo.

Entrada

A primeira linha contém um inteiro N , o tamanho do tabuleiro do jogo, que é quadrado. Cada uma das N linhas seguintes contém N caracteres, que podem ser (aspas para melhor clareza):

- ‘.’ um espaço vazio;
- ‘o’ uma comida;
- ‘A’ um fantasma.

Saída

Seu programa deve produzir uma única linha contendo um único inteiro, a quantidade máxima de comida que o Pacman pode levar para casa.

Restrições

- $2 \leq N \leq 100$
- Não há um fantasma e uma comida na mesma posição.
- Não há fantasma nem comida na posição inicial do Pacman (ou seja, o primeiro caractere da primeira linha do tabuleiro é ‘.’).

Exemplos

Entrada	Saída
5 .ooo. .ooA .Aoo Aoooo .ooo	6

Entrada	Saída
3 .o. oAA ooo	4

Fechadura

Nome do arquivo fonte: `fechadura.c`, `fechadura.cpp`, `fechadura.pas`, `fechadura.java`, ou `fechadura.py`

Joãozinho estava um dia chegando em casa quando percebeu que havia perdido a chave da porta. Desesperado, ele resolveu pedir ajuda a seu amigo Roberto, que em poucos segundos conseguiu abrir a porta usando suas ferramentas.

Admirado com a velocidade em que seu amigo conseguiu abrir a porta de sua casa sem a chave, ele decidiu perguntar como ele tinha conseguido aquilo. Roberto explicou que a fechadura da casa de Joãozinho é baseada em um sistema de pinos de tamanhos diferentes que, uma vez alinhados na mesma altura M , possibilitam a abertura da porta.

Uma fechadura é um conjunto de N pinos dispostos horizontalmente que podem ser movimentados para cima ou para baixo com o auxílio de uma chave de metal que, ao ser inserida dentro da fechadura, pode aumentar ou diminuir em 1mm, simultaneamente, a altura de quaisquer dois pinos consecutivos.

Joãozinho como um exemplar perfeccionista decidiu desbloquear sua fechadura na menor quantidade de movimentos, onde cada movimento consiste em escolher dois pinos consecutivos da fechadura e aumentar ou diminuir a altura dos dois pinos em 1mm. Após todos os pinos possuírem altura exatamente igual a M , a fechadura é desbloqueada.

Entrada

A primeira linha da entrada contém dois inteiros N e M representando, respectivamente, a quantidade de pinos da fechadura e a altura em que eles devem ficar para a fechadura ser desbloqueada.

A segunda linha da entrada contém N inteiros, representando as alturas dos pinos da fechadura.

Saída

Seu programa deve imprimir uma linha contendo um inteiro representando a quantidade mínima de movimentos para desbloquear a fechadura.

Restrições

- $1 \leq N \leq 1000$
- $1 \leq M \leq 100$
- Cada altura dos pinos está entre 1 e 100.
- É garantido que os casos de testes sempre possuem uma solução.

Exemplos

Entrada	Saída
4 50 45 45 55 55	10

Entrada	Saída
5 84 84 39 17 72 94	77

Matriz Escada

Nome do arquivo fonte: `escada.c`, `escada.cpp`, `escada.pas`, `escada.java`, ou `escada.py`

Joãozinho está aprendendo sobre matrizes. Hoje ele aprendeu como deixar matrizes na forma escada, e está exercitando. Para ajudá-lo, você deve escrever um programa que determine se o resultado dele realmente está no formato correto.

Uma matriz está na forma escada quando, para cada linha, as condições a seguir forem satisfeitas:

- Se a linha só possuir zeros, então todas as linhas abaixo desta também só possuem zeros.
- Caso contrário, seja X o elemento diferente de zero mais à esquerda da linha; então, para todas as linhas abaixo da linha de X , todos os elementos nas colunas à esquerda de X e na coluna de X são iguais a zero.

Entrada

A primeira linha possui dois inteiros N e M , as dimensões da matriz. Cada uma das N linhas seguintes contém M inteiros não-negativos, os elementos da matriz.

Saída

Seu programa deve produzir uma única linha, contendo o caractere ‘S’ caso a matriz esteja no formato escada, ou ‘N’, caso contrário.

Restrições

- $1 \leq N \leq 500$ e $1 \leq M \leq 500$.
- Cada elemento da matriz está entre 0 e 10^5 .

Informações sobre a pontuação

- Em um conjunto de casos de teste equivalente a 80 pontos, $N \leq 50$ e $M \leq 50$.

Exemplos

Entrada	Saída
<pre>4 6 1 2 9 9 9 9 0 0 3 9 9 9 0 0 0 0 5 9 0 0 0 0 0 6</pre>	S

Entrada	Saída
<pre>5 8 0 5 1 0 3 2 2 0 0 0 0 0 4 0 1 2 0 0 0 0 0 0 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</pre>	S

Entrada	Saída
5 5	N
1 1 2 3 4	
0 1 1 4 5	
0 1 2 3 6	
0 0 0 2 0	
0 0 0 0 0	