Maratona Interna de Programação

5 de Setembro, 2015

Caderno de Problemas

Informações Gerais

Este caderno contém 6 problemas. As páginas estão numeradas de 1 a 8, não contando esta página de rosto. Verifique se o caderno está completo.

A) Sobre a entrada

1. A entrada de seu programa deve ser lida da entrada padrão.
2. A entrada é composta de um único ou vários casos de teste, descrito em um número de linhas que depende do problema.
3. Quando uma linha da entrada contém vários valores, estes são separados por um único espaço em branco; a entrada não contém nenhum outro espaço em branco.
4. Cada linha, incluindo a última, contém exatamente um caractere final-de-linha.
5. O final da entrada coincide com o final do arquivo.

B) Sobre a saída

1. A saída de seu programa deve ser escrita na saída padrão.
2. Quando uma linha da saída contém vários valores, estes devem ser separados por um único espaço em branco; a saída não deve conter nenhum outro espaço em branco.
3. Cada linha, incluindo a última, deve conter exatamente um caractere final-de-linha.

Problema A
Peça Perdida

Arquivo: `peca.[c|cpp|java]`

Joãozinho adora quebra-cabeças, essa é sua brincadeira favorita. O grande problema, porém, é que às vezes o jogo vem com uma peça faltando. Isso irrita bastante o pobre menino, que tem de descobrir qual peça está faltando e solicitar uma peça de reposição ao fabricante do jogo. Sabendo que o quebra-cabeças tem N peças, numeradas de 1 a N e que exatamente uma está faltando, ajude Joãozinho a saber qual peça ele tem de pedir.

Escreva um programa que, dado um inteiro N e $N-1$ inteiros numerados de 1 a N, descubra qual inteiro está faltando.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). A entrada contém 2 linhas. A primeira linha contém um inteiro $N(2 \leq N \leq 1.000)$. A segunda linha contém $N-1$ inteiros numerados de 1 a N (sem repetições).

Saída

Seu programa deve imprimir, na saída padrão, uma única linha, contendo o número que está faltando na sequência dada.

Exemplo de entrada	**Exemplo de saída**
3 3 1	2

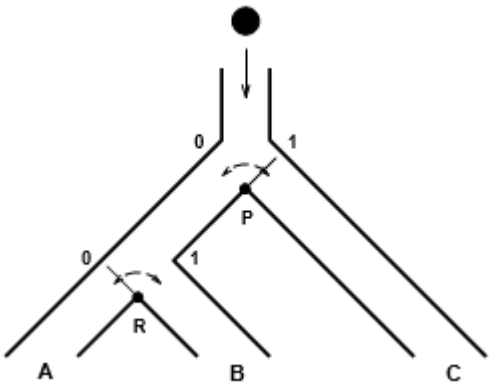
Exemplo de entrada	**Exemplo de saída**
5 1 2 3 5	4

Exemplo de entrada	**Exemplo de saída**
4 2 4 3	1

Problema B
Flíper

Arquivo: `fliper.[c|cpp|java]`

Flíper é um tipo de jogo onde uma bolinha de metal cai por um labirinto de caminhos até chegar na parte de baixo do labirinto. A quantidade de pontos que o jogador ganha depende do caminho que a bolinha seguir. O jogador pode controlar o percurso da bolinha mudando a posição de algumas portinhas do labirinto. Cada portinha pode estar na posição 0, que significa virada para a esquerda, ou na posição 1 que quer dizer virada para a direita. Considere o flíper da figura abaixo, que tem duas portinhas. A portinha P está na posição 1 e a portinha R, na posição 0. Desse jeito, a bolinha vai cair pelo caminho B.



Você deve escrever um programa que, dadas as posições das portinhas P e R, neste flíper da figura, diga por qual dos três caminhos, A, B ou C, a bolinha vai cair!

Entrada

A entrada é composta por apenas uma linha contendo dois números P e R, indicando as posições das duas portinhas do flíper da figura.

Saída

A saída do seu programa deve ser também apenas uma linha, contendo uma letra maiúscula que indica o caminho por onde a bolinha vai cair: `'A'`, `''B'` ou `'C'`.

Restrições

- O número P pode ser 0 ou 1. O número R pode ser 0 ou 1.

Exemplo de entrada	**Exemplo de saída**
1 0	B

Exemplo de entrada	**Exemplo de saída**
0 0	C

Problema C
Notas

Arquivo: `notas.[c|cpp|java]`

O professor Arquimedes precisa da sua ajuda para descobrir qual é a nota mais frequente entre as notas que os alunos dele tiraram na última prova. A turma tem N alunos e seu programa deve imprimir a nota que aparece mais vezes na lista de N notas. Se houver mais de uma nota mais frequente, você deve imprimir a maior delas! Por exemplo, se a turma tiver $N = 10$ alunos e as notas forem [20, 25, 85, 40, 25, 90, 25, 40, 55, 40], as notas mais frequentes são 25 e 40, ocorrendo três vezes cada. Seu programa, então, deve imprimir 40.

Entrada

A entrada consiste de duas linhas. A primeira linha contém um número inteiro N , o número de alunos na turma. A segunda linha contém N inteiros, que é a lista de notas dos alunos.

Saída

Seu programa deve imprimir apenas uma linha contendo apenas um número, a nota mais frequente da lista.

Restrições

- $1 \leq N \leq 200$
- O valor de todas as notas é um inteiro entre 0 e 100, inclusive.

Exemplo de entrada	**Exemplo de saída**
10 20 25 85 40 25 90 25 40 55 40	40

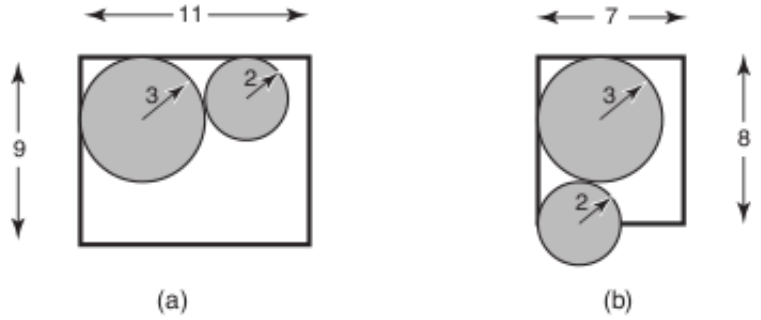
Exemplo de entrada	**Exemplo de saída**
12 45 0 33 70 12 55 70 70 90 55 70 100	70

Problema D
Elevador

Arquivo: `elevador.[c|cpp|java]`

A FCC (Fábrica de Cilindros de Carbono) fabrica vários tipos de cilindros de carbono. A FCC está instalada no décimo andar de um prédio, e utiliza os vários elevadores do prédio para transportar os cilindros. Por questão de segurança, os cilindros devem ser transportados na posição vertical; como são pesados, no máximo dois cilindros podem ser transportados em uma única viagem de elevador. Os elevadores têm formato de paralelepípedo e sempre têm altura maior que a altura dos cilindros.

Para minimizar o número de viagens de elevador para transportar os cilindros, a FCC quer, sempre que possível, colocar dois cilindros no elevador. A figura abaixo ilustra, esquematicamente (vista superior), um caso em que isto é possível (a), e um caso em que isto não é possível (b):



Como existe uma quantidade muito grande de elevadores e de tipos de cilindros, a FCC quer que você escreva um programa que, dadas as dimensões do elevador e dos dois cilindros, determine se é possível colocar os dois cilindros no elevador.

Entrada

A entrada contém vários casos de teste. A primeira e única linha de cada caso de teste contém quatro números inteiros L, C, R_1 e R_2, separados por espaços em branco, indicando respectivamente a largura do elevador ($1 \leq L \leq 100$), o comprimento do elevador ($1 \leq C \leq 100$), e os raios dos cilindros ($1 \leq R_1, R_2 \leq 100$).

O último caso de teste é seguido por uma linha que contém quatro zeros separados por espaços em branco

Saída

Para cada caso de teste, o seu programa deve imprimir uma única linha com um único caractere: `'S'` se for possível colocar os dois cilindros no elevador e `'N'` caso contrário.

Entrada	**Saída**
11 9 2 3	S
7 8 3 2	N
10 15 3 7	N
8 9 3 2	S
0 0 0 0	

Problema E
PacMan

Arquivo: `pacmano.[c|cpp|java]`

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

- $2 \leq N \leq 10$
- 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 `'.'`).

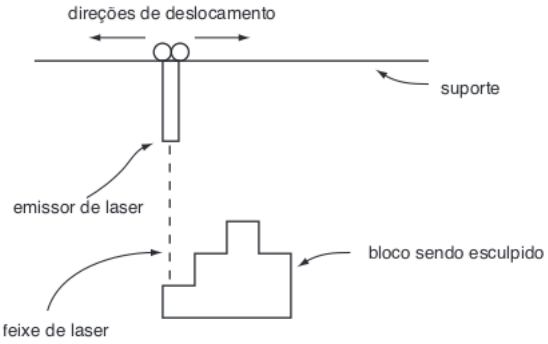
Entrada	**Saída**
5 .ooo. ..ooA ..Aoo Aoooo ..ooo	6

Entrada	**Saída**
3 .o. oAA ooo	4

Problema F
Escultura a Laser

Arquivo: `laser.[c|cpp|java]`

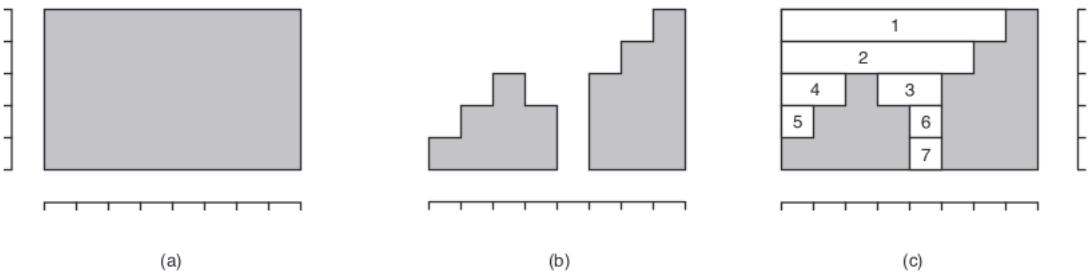
Desde a sua invenção, em 1958, os raios laser têm sido utilizados em uma imensa variedade de aplicações, como equipamentos eletrônicos, instrumentos cirúrgicos, armamentos, e muito mais.



A figura acima mostra um diagrama esquemático de um equipamento para esculpir, com laser, um bloco de material maciço. Na figura vemos um emissor laser que se desloca horizontalmente para a direita e para a esquerda com velocidade constante. Quando o emissor é ligado durante o deslocamento, uma camada de espessura constante é removida do bloco, sendo vaporizada pelo laser.

A figura abaixo ilustra o processo de escultura a laser, mostrando um exemplo de (a) um bloco, com 5 mm de altura por 8 mm de comprimento, no início do processo, (b) o formato que se deseja que o bloco esculpido tenha, e (c) a sequência de remoção das camadas do bloco durante o processo, considerando que a cada varredura uma camada de espessura de 1 mm é removida.

Na primeira varredura, o pedaço numerado como 1 é removido; na segunda varredura, o pedaço numerado como 2 é removido, e assim por diante. Durante o processo de remoção, o laser foi ligado um total de 7 vezes, uma vez para cada pedaço de bloco removido.



Escreva um programa que, dados a altura do bloco, o comprimento do bloco, e a forma final que o bloco deve ter, determine o número total vezes de que o laser deve ser ligado para esculpir o bloco.

Entrada

A entrada contém vários casos de teste. Cada caso de teste é composto por duas linhas. A primeira linha de um caso de teste contém dois números inteiros A e C, separados por um espaço em branco, indicando respectivamente a altura ($1 \leq A \leq 10^4$) e o comprimento ($1 \leq C \leq 10^4$)

do bloco a ser esculpido, em milímetros. A segunda linha contém C números inteiros X_i, cada um indicando a altura final, em milímetros, do bloco entre as posições i e $i + 1$ ao longo do comprimento ($0 \leq X_i \leq A$, para $0 \leq i \leq C - 1$). Considere que a cada varredura uma camada de espessura 1 milímetro é removida do bloco ao longo dos pontos onde o laser está ligado.

O final da entrada é indicado por uma linha que contém apenas dois zeros, separados por um espaço em branco.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha, contendo um numero inteiro, indicando o número de vezes que o laser deve ser ligado para esculpir o bloco na forma indicada.

Entrada	**Saída**
5 8 1 2 3 2 0 3 4 5 3 3 1 0 2 4 3 4 4 1 0 0	7 3 3