OBI2017

Caderno de Tarefas

Modalidade **Programação** • **Nível 2** • Fase **1**

12 de maio de 2017

A PROVA TEM DURAÇÃO DE **2** HORAS

Promoção:

Sociedade Brasileira de Computação

Apoio:

v1.1

Instruções

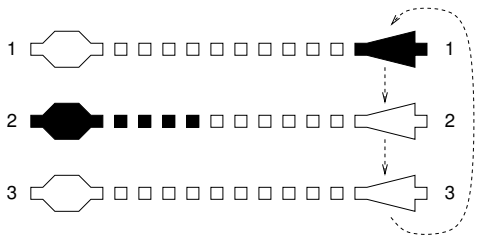
LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 6 páginas (não contando a folha de rosto), numeradas de 1 a 6. 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; soluções na linguagem Python devem ser arquivos com sufixo *.py2* para python2 e .py3 para python3; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*. 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 pen-drive, 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,input,print,write*
  - em Javascript: *scanf, printf*
- 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.

Game-10

Nome do arquivo: `game10.c`, `game10.cpp`, `game10.pas`, `game10.java`, `game10.js` *ou* `game10.py`

No princípio dos anos 1980 surgiram nos colégios os primeiros relógios de pulso digitais com joguinhos. Era uma febre entre os alunos e quem tinha um era muito popular na hora do recreio. Os joguinhos eram bem simples, mas muito legais. Um dos primeiros era o Game-10, no qual você controlava um avião que aparecia na parte direita do visor. Na parte esquerda aparecia um disco voador em qualquer uma de três posições, aleatoriamente, e lançava um míssil. O objetivo do jogador era movimentar o avião verticalmente para que ficasse na frente do disco voador (na mesma linha horizontal, do lado direito) e atirar para interceptar o míssil antes que esse atingisse o avião.



Como o movimento do avião era feito com apenas um botão, só dava para movimentar em um sentido: ao apertar o botão sucessivas vezes, o avião se movia na sequência de posições $\cdots 1 \to 2 \to 3 \to 1 \to 2 \to 3 \to 1 \cdots$ Veja que, na situação da figura, o jogador deveria apertar o botão apenas uma vez, para ir da posição 1 para a posição 2, e conseguir atirar e interceptar o míssil.

Neste problema vamos considerar que existem N posições e não apenas três. Dado o número de posições N, a posição D na qual o disco voador aparece, e a posição A onde está o avião, seu programa deve computar o número mínimo de vezes que o jogador precisa apertar o botão para movimentar o avião até a mesma posição do disco voador e poder atirar!

Entrada

A primeira linha da entrada contém um inteiro N, o número de posições. A segunda linha contém um inteiro D, a posição do disco voador. A terceira linha contém um inteiro A, a posição do avião.

Saída

Seu programa deve imprimir uma linha contendo um inteiro, o número mínimo de vezes que o jogador deve apertar o botão para poder atirar.

Restrições

- $3 \leq N \leq 100$
- $1 \leq D, A \leq N$

Exemplos

Entrada	**Saída**
3 2 1	1

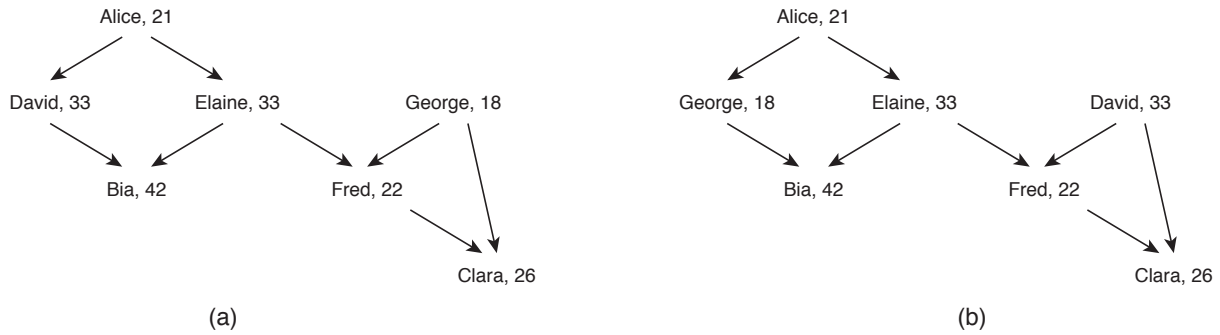
Entrada	**Saída**
20 8 13	15

O Chefe

Nome do arquivo: `chefe.c`, `chefe.cpp`, `chefe.pas`, `chefe.java`, `chefe.js` *ou* `chefe.py`

Todos conhecem Iks, a última moda em redes sociais, que fez tanto sucesso que competidores como Facebook e Google+ estão começando a ter dificuldades financeiras. Assim como muitas companhias ".com", Iks surgiu em uma pequena garagem, mas hoje emprega milhares de pessoas no mundo todo.

O sistema de gerência utilizado em Iks é bem diferente do padrão. Por exemplo, não há diretorias ou superintendências. No entanto, como é usual em outras companhias, há uma cadeia (ou melhor, várias cadeias) de comando: uma pessoa pode gerenciar outras pessoas, e pode ser gerenciada por outras pessoas. As figuras abaixo mostram a cadeia de comando para alguns empregados, junto com suas idades.



Uma pessoa P_1 pode gerenciar outra pessoa P_2 diretamente (quando P_1 é o superior imediato de P_2) ou indiretamente (quando P_1 gerencia diretamente uma pessoa P_3 que gerencia P_2 direta ou indiretamente). Por exemplo, na figura (a) acima, Alice gerencia David diretamente e Clara indiretamente. Uma pessoa não gerencia a si própria, nem direta nem indiretamente.

Um folclore que apareceu em Wall Street é que Iks é tão bem sucedido porque em sua rede de comando um(a) gerente é sempre mais jovem do que as pessoas que ele(a) gerencia. Como podemos ver na figura acima, isso não é verdade. Mas esse folclore incentivou Iks a desenvolver uma ferramenta para analisar o seu sistema de gerenciamento, e estudar se tem alguma influência no sucesso da empresa. Você foi contratado para trabalhar nessa ferramenta.

Dadas a descrição da cadeia de comando na Iks e as idades de seus empregados, escreva um programa que execute uma série de instruções. Instruções podem ser de dois tipos: trocas de gerência e perguntas. Uma instrução de troca de gerência faz dois empregados A e B trocarem suas posições na cadeia de comando. Como exemplo, a figura (b) acima mostra a cadeia de comando resultante quando David e George trocam suas respectivas posições na cadeia de comando. Uma instrução de pergunta identifica um empregado A e deseja saber a idade do mais jovem gerente (direto ou indireto) de A na cadeia de comando. Por exemplo, no cenário da figura (a) acima a idade do(a) gerente mais jovem de Clara é 18 anos; já no cenário da figura (b), a idade do(a) gerente mais jovem de Clara é 21 anos.

Entrada

A entrada é composta de várias linhas. A primeira linha contém três inteiros N, M e I, indicando respectivamente o número de empregados, o número de relações de gerência direta e o número de instruções. Empregados são identificados por números de 1 a N. A segunda linha contém N inteiros K_i, onde K_i indica a idade do empregado de número i.

Cada uma das M linhas seguintes contém dois inteiros X e Y, indicando que X gerencia Y diretamente. Seguem-se I linhas, cada uma descrevendo uma instrução. Uma instrução de troca de gerência é descrita em uma linha contendo o identificador `T` seguido de dois inteiros A e B, indicando os dois empregados que devem trocar seus lugares na cadeia de comando. Uma instrução de pergunta é descrita em uma linha contendo o identificador `P` seguido de um inteiro E , indicando um empregado. A última instrução será sempre do tipo pergunta.

Saída

Para cada instrução de pergunta seu programa deve imprimir uma linha contendo um único inteiro, a idade da pessoa mais jovem que gerencia (direta ou indiretamente) o empregado nomeado na pergunta. Se o empregado nomeado não possui um gerente, imprima o caractere '`*`' (asterisco).

Restrições

- $1 \leq N \leq 500$
- $0 \leq M \leq 60 \times 10^3$
- $1 \leq I \leq 500$
- $1 \leq K_i \leq 100$, para $1 \leq i \leq N$
- $1 \leq X, Y \leq N$, $X \neq Y$
- $1 \leq A, B \leq N$
- $1 \leq E \leq N$

Exemplos

Entrada	**Saída**
7 8 9 21 33 33 18 42 22 26 1 2 1 3 2 5 3 5 3 6 4 6 4 7 6 7 P 7 T 4 2 P 7 P 5 T 1 4 P 7 T 4 7 P 2 P 6	18 21 18 18 * 26

Entrada	**Saída**
6 5 6 10 20 30 40 50 60 1 5 1 4 3 6 2 5 4 5 P 1 P 5 P 6 T 1 6 P 1 P 4	* 10 30 30 60

Botas Trocadas

Nome do arquivo: `botas.c`, `botas.cpp`, `botas.pas`, `botas.java`, `botas.js` *ou* `botas.py`

A divisão de Suprimentos de Botas e Calçados do Exército comprou um grande número de pares de botas de vários tamanhos para seus soldados. No entanto, por uma falha de empacotamento da fábrica contratada, nem todas as caixas entregues continham um par de botas correto, com duas botas do mesmo tamanho, uma para cada pé. O sargento mandou que os recrutas retirassem todas as botas de todas as caixas para reembalá-las, desta vez corretamente.

Quando o sargento descobriu que você sabia programar, ele solicitou com a gentileza habitual que você escrevesse um programa que, dada a lista contendo a descrição de cada bota entregue, determina quantos pares corretos de botas poderão ser formados no total.

Entrada

A primeira linha da entrada contém um inteiro N indicando o número de botas individuais entregues. Cada uma das N linhas seguintes descreve uma bota, contendo um número inteiro M e uma letra L, separados por um espaço em branco. M indica o número do tamanho da bota e L indica o pé da bota: $L =$ '`D`' indica que a bota é para o pé direito, $L =$ '`E`' indica que a bota é para o pé esquerdo.

Saída

Seu programa deve imprimir uma única linha contendo um único número inteiro indicando o número total de pares corretos de botas que podem ser formados.

Restrições

- $2 \leq N \leq 10^4$
- N é par
- $30 \leq M \leq 60$
- L é o caractere '`D`' ou o caractere '`E`'

Exemplos

Entrada	**Saída**
4 40 D 41 E 41 D 40 E	2

Entrada	**Saída**
6 38 E 39 E 40 D 38 D 40 D 37 E	1