



OBI2010

Caderno de Tarefas

Modalidade Programação • Nível 1, Fase 2

8 de maio de 2010

A PROVA TEM DURAÇÃO DE 4 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 8 páginas (não contando esta folha de rosto), numeradas de 1 a 8. 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*. 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*.
- 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.

Sedex Marciano

Nome do arquivo fonte: `marciano.c`, `marciano.cpp`, ou `marciano.pas`

Estamos no ano 2048 e um dos sonhos da humanidade torna-se finalmente realidade: a colonização do planeta Marte. Nossos primeiros colonizadores acabam de chegar, e já começam a fazer as preparações (como a instalação de cúpulas de oxigênio e tratamento do solo para agricultura) para que mais pessoas possam tentar uma nova vida no planeta vizinho.

Apesar dos avanços tecnológicos e desafios vencidos, ainda resta um grande problema: os foguetes usados para ir a Marte ainda são complicados e caros. Com isso, fica difícil enviar suprimentos para os nossos colonos (enquanto a agricultura ainda não é possível) por muito tempo. Assim, a agência espacial contratou o SBC (Serviço Balístico Cósmico), que desenvolveu um canhão super-potente que consegue disparar esferas até Marte, sem precisar gastar milhões de dólares em equipamento e combustível.

Agora, tudo o que é necessário fazer para enviar suprimentos a Marte é colocar uma caixa com as encomendas dentro de uma esfera e disparar a mesma até seu destino.

Tarefa

Dadas as dimensões de uma caixa com suprimentos e o raio interno da esfera que é disparada pelo canhão, seu programa deverá dizer se é possível enviar tal caixa para Marte usando tal esfera.

Entrada

Cada entrada contém apenas uma linha com quatro inteiros L , A , P e R , ($0 \leq L, A, P, R \leq 1000$) que representam, respectivamente, a largura, altura e profundidade da caixa, e o raio da esfera.

Saída

Seu programa deve imprimir um único caractere: 'S' (sem aspas) se é possível colocar a caixa dentro da esfera, ou 'N' (sem aspas) caso contrário.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 20 pontos, $0 \leq L, A, P, R \leq 20$.

Exemplos

Entrada 10 20 30 30	Saída S
Entrada 10 10 10 7	Saída N
Entrada 2 4 4 3	Saída S

Fusões

Nome do arquivo fonte: `fusoes.c`, `fusoes.cpp`, ou `fusoes.pas`

A informatização dos sistemas bancários permitiu grandes economias de tempo e dinheiro, permitindo que vários tipos de transações financeiras pudessem ser realizadas pela Internet. Para possibilitar isso, cada banco recebeu um *código bancário*, que é um número utilizado pelos sistemas de computador para identificar cada banco.

Quando um banco decide comprar outro, ocorre o que se chama uma *fusão* : os dois bancos tornam-se um só banco. Para manter compatibilidade com os sistemas eletrônicos dos bancos, qualquer um dos códigos dos antigos bancos pode ser usado para se referir ao novo banco.

Com a crise econômica internacional, as fusões entre bancos têm sido cada vez mais comuns; por isso, muitas vezes é difícil decidir se dois códigos bancários na realidade se referem ao mesmo banco (devido aos dois bancos terem se fundido, diretamente ou não).

Tarefa

Escreva um programa que, dada uma série de fusões entre bancos, responde a várias consultas perguntando se dois códigos bancários se referem ao mesmo banco.

Entrada

A primeira linha da entrada contém dois inteiros N e K , indicando o número de bancos e o número de operações efetuadas ($1 \leq N \leq 100.000$, $1 \leq K \leq 100.000$). Os códigos de cada um dos N bancos, inicialmente, são os inteiros de 1 até N .

Cada uma das K linhas seguintes descreve ou uma fusão entre bancos ou uma consulta.

- Uma fusão é descrita na entrada como uma linha que começa com o caractere ‘F’, um espaço, e dois códigos bancários, que se referem aos dois bancos que estão sofrendo a fusão, separados por um espaço em branco;
- Uma consulta é descrita na entrada como uma linha que começa com o caractere ‘C’, um espaço, e os dois códigos a serem consultados, separados por um espaço em branco. Os códigos bancários consultados são sempre distintos.

As fusões são sempre realizadas entre bancos diferentes, e todos os códigos bancários fornecidos na entrada são válidos.

Saída

Seu programa deve imprimir uma linha para cada consulta na entrada. Caso os dois códigos bancários consultados se refiram ao mesmo banco, imprima uma linha contendo o caractere ‘S’; caso contrário, imprima uma linha contendo apenas o caractere ‘N’.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 40 pontos, $N \leq 100$ e $K \leq 100$.

Exemplos

Entrada	Saída
3 5 C 1 2 F 1 2 C 1 2 F 1 3 C 1 3	N S S

Entrada	Saída
4 5 F 1 2 F 2 3 C 1 3 F 2 4 C 1 4	S S

Entrada	Saída
4 4 F 1 2 F 3 4 F 1 3 C 2 4	S

Lista de Chamada

Nome do arquivo fonte: chamada.c, chamada.cpp, ou chamada.pas

Tia Joana é uma respeitada professora e tem vários alunos. Em sua última aula, ela prometeu que iria sortear um aluno para ganhar um bônus especial na nota final: ela colocou N pedaços de papel numerados de 1 a N em um saquinho e sorteu um determinado número K ; o aluno premiado foi o K -ésimo aluno na lista de chamada.

O problema é que a Tia Joana esqueceu o diário de classe, então ela não tem como saber qual número corresponde a qual aluno. Ela sabe os nomes de todos os alunos, e que os números deles, de 1 até N , são atribuídos de acordo com a ordem alfabética, mas os alunos dela estão muito ansiosos e querem logo saber quem foi o vencedor.

Tarefa

Dado os nomes dos alunos da Tia Joana e o número sorteado, determine o nome do aluno que deve receber o bônus.

Entrada

A primeira linha contém dois inteiros N e K separados por um espaço em branco ($1 \leq K \leq N \leq 100$). Cada uma das N linhas seguintes contém uma cadeia de caracteres de tamanho mínimo 1 e máximo 20 representando os nomes dos alunos. Os nomes são compostos apenas por letras minúsculas de ‘a’ a ‘z’.

Saída

Seu programa deve imprimir uma única linha, contendo o nome do aluno que deve receber o bônus.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos, $N \leq 3$.

Exemplos

Entrada	Saída
5 1 maria joao carlos vanessa jose	carlos

Entrada	Saída
5 5 maria joao carlos vanessa jose	vanessa

Entrada	Saída
5 3 maria joao carlos vanessa jose	jose

Dança Indígena

Nome do arquivo fonte: `danca.c`, `danca.cpp`, ou `danca.pas`

A OBI (Organização Brasileira dos Índios) promoverá um festival indígena, onde várias tribos irão se reunir e fazer demonstrações de cultura, como artesanato, danças, pinturas, comidas e etc.

Uma das tribos é a dos Tunak Tunak, que possuem uma apresentação de dança muito peculiar. Nessa dança, existem N toras de madeira encrustadas no chão, dispostas de maneira circular e igualmente espaçadas. Em algumas dessas toras fica um índio, olhando em sentido horário ou anti-horário.

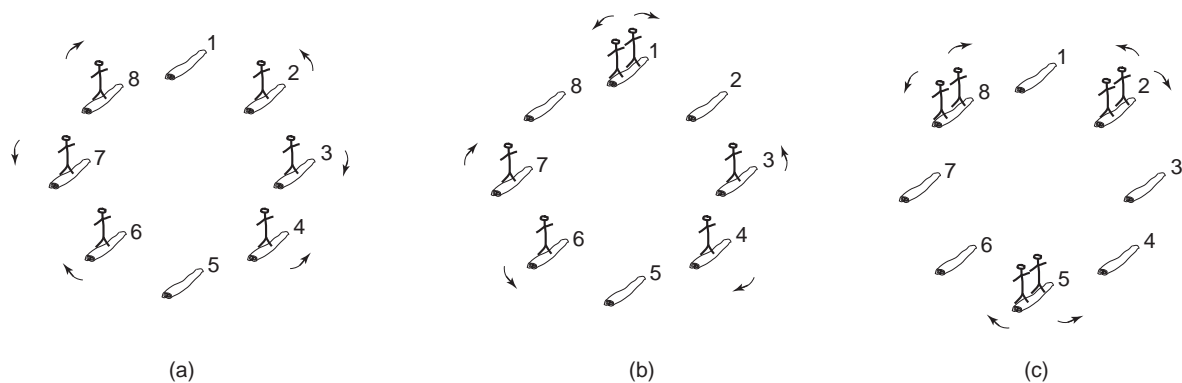
A cada batida do tambor, os índios pulam para a próxima tora (que depende da direção para onde ele está olhando no momento). Durante a dança, porém, algumas coisas podem acontecer:

- Dois índios que pulam em sentidos opostos caem na mesma tora ao mesmo tempo. Nesse caso, ambos permanecem nas toras, mas passam a pular na direção contrária a partir da próxima batida de tambor (isso é, quem estava pulando em sentido horário passa a pular em sentido anti-horário, e vice-versa)
- Dois índios em toras consecutivas vão pular um em direção ao outro. Nesse caso, os índios simplesmente não pulam (para não causar nenhum acidente), e, assim como no caso anterior, passam a pular no sentido contrário a partir da próxima batida de tambor.

Note que se o índio não pula e inverte seu sentido, mas ao mesmo tempo um outro índio cair na mesma tora no sentido contrário, caímos no primeiro caso, e ambos os índios na tora invertem seus sentidos (assim, o índio que estava na tora anteriormente inverte seu sentido novamente).

A dança termina quando as toras ocupadas por um índio são exatamente as mesmas toras ocupadas no início da dança, não importando qual índio está em cada tora e nem os sentidos para onde eles estão pulando.

A figura abaixo ilustra (a) uma configuração inicial com oito toras e seis índios; (b) a posição dos índios após uma batida de tambor; e (c) a posição dos índios após duas batidas de tambor.



Tarefa

Os índios querem se preparar para a dança e precisam saber quanto tempo ela vai durar.

Para isso, você deverá escrever um programa que, dados a quantidade de toras que serão utilizadas, a quantidade de índios e a posição inicial de cada um, calcule quantas batidas de tambor levará para que a dança termine.

Entrada

A primeira linha da entrada possui 2 inteiros: N ($3 \leq N \leq 1.000.000$) e E ($1 \leq E \leq 1000$), que são, respectivamente, a quantidade de toras e a quantidade de índios que irão dançar ($E \leq N$). As próximas E

linhas contém, cada uma, a descrição da posição inicial de cada índio. Cada linha possui dois inteiros: V ($1 \leq V \leq N$) e D ($D = 1$ ou $D = -1$) que representam, respectivamente, o número da tora onde o índio inicia e seu sentido inicial (1 se horário, -1 se anti-horário). A numeração das toras obedece o sentido horário. No início da dança uma tora terá, no máximo, um índio.

Saída

Seu programa deverá exibir um número inteiro representando a quantidade de batidas de tambor necessárias para que a dança acabe.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 40 pontos, $N \leq 100$ e $E \leq 100$.

Exemplos

Entrada	Saída
6 4 2 1 3 1 5 1 6 1	3

Entrada	Saída
3 1 2 -1	3

Entrada	Saída
8 6 2 -1 3 1 4 -1 6 1 7 -1 8 1	4