9^o Treino para Alunos da UFPR

6 de Julho de 2013

Sevidor BOCA:
`http://maratona.c3sl.ufpr.br/boca/`

Organizadores:
Vinicius Kwiecien Ruoso e Ricardo Tavares de Oliveira

Lembretes:

- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova.
- A correção é automatizada, portanto, siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa. Deve-se considerar entradas e saídas padrão.
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. As soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas.
- Teste seu programa antes de submetê-lo. A cada problema detectado (erro de compilação, erro em tempo de execução, solução incorreta, formatação imprecisa, tempo excedido ...), há penalização de 20 minutos. O tempo é critério de desempate entre duas ou mais equipes com a mesma quantidade de problemas resolvidos.
- Utilize o *clarification* para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos.

A: Bolhas e Baldes

`File: bolhas.[c|cpp|java|pas]`

Andrea, Carlos e Marcelo são muito amigos e passam todos os finais de semana à beira da piscina. Enquanto Andrea se bronzeia ao sol, os dois ficam jogando Bolhas. Andrea, uma cientista da computação muito esperta, já disse a eles que não entende por que passam tanto tempo jogando um jogo tão primário.

Usando o computador portátil dela, os dois geram um inteiro aleatório N e uma seqüência de inteiros, também aleatória, que é uma permutação de $1, 2, ..., N$.

O jogo então começa, cada jogador faz um movimento, e a jogada passa para o outro jogador. Marcelo é sempre o primeiro a começar a jogar.

Um movimento de um jogador consiste na escolha de um par de elementos consecutivos da seqüência que estejam fora de ordem e em inverter a ordem dos dois elementos. Por exemplo, dada a seqüência $1, 5, 3, 4, 2$, o jogador pode inverter as posições de 5 e 3 ou de 4 e 2, mas não pode inverter as posiçõoes de 3 e 4, nem de 5 e 2. Continuando com o exemplo, se o jogador decide inverter as posições de 5 e 3 então a nova seqüência será $1, 3, 5, 4, 2$.

Mais cedo ou mais tarde, a seqüência ficará ordenada. Perde o jogador impossibilitado de fazer um movimento.

Andrea, com algum desdém, sempre diz que seria mais simples jogar cara ou coroa, com o mesmo efeito. Sua missão, caso decida aceitá-la, é determinar quem ganha o jogo, dada a seqüência inicial.

Entrada

A entrada contém vários casos de teste. Os dados de cada caso de teste estão numa única linha, e são inteiros separados por um espaço em branco. Cada linha contém um inteiro N, $2 \leq N \leq 10^5$, seguido da seqüência inicial $P = (X_1, X_2, ..., X_N)$ de N inteiros distintos dois a dois, onde $1 \leq X_i \leq N$ para $1 \leq i \leq N$.

O final da entrada é indicado por uma linha que contém apenas o número zero.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha, com o nome do vencedor, igual a *Carlos* ou a *Marcelo*, sem espaços em branco.

Exemplo de entrada	**Exemplo de saída**

`5 1 5 3 4 2`
`5 5 1 3 4 2`
`5 1 2 3 4 5`
`6 3 5 2 1 4 6`
`5 5 4 3 2 1`
`6 6 5 4 3 2 1`
`0`
	`Marcelo`
	`Carlos`
	`Carlos`
	`Carlos`
	`Carlos`
	`Marcelo`

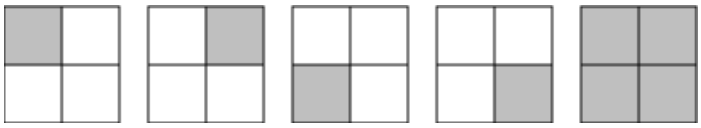
B: Feynman

`File: feynman.[c|cpp|java|pas]`

Richard Phillips Feynman era uma físico americando muito famoso e ganhador do Prêmio Nobel de Física. Ele trabalhava em física teórica e também foi pioneiro no campo da computação quântica. Ele visitou a América do Sul por dez meses, dando palestras e aproveitando a vida nos trópicos. Ele também é conhecido pelos livros *"Surely You're Joking, Mr. Feynman!"* e *"What Do You Care What Other People Think?"*, que inclui algumas de suas aventuras abaixo do equador.

Sua paixão da vida inteira era resolver e criar quebra-cabeças, trancas e códigos. Recentemente, um fazendeiro idoso da América do Sul, que hospedou o jovem físico em 1949, achou alguns papéis e notas que acredita-se terem pertencido a Feynman. Entre anotações sobre mesóns e eletromagnetismo, havia um guardanapo onde ele escreveu um simples desafio: "quantos quadrados diferentes existem em um quadriculado de $N \times N$ quadrados?".

No mesmo guardanapo havia um desenho, que está reproduzido abaixo, mostrando que para $N = 2$, a resposta é 5.



Entrada

A entrada contém diversos casos de teste. Cada caso de teste é composto de uma única linha, contendo apenas um inteiro N, representando o número de quadrados em cada lado do quadriculado ($1 \leq N \leq 100$).

O final da entrada é indicado por uma linha contendo apenas um zero.

Saída

Para cada caso de teste na entrada, seu programa deve imprimir uma única linha, contendo o número de diferentes quadrados para a entrada correspondente.

Exemplo de entrada	**Exemplo de saída**
2 1 8 0	5 1 204

C: Heart Piece!

File: heart.[c|cpp|java|pas]

Felipe é um grande fã da série de jogos *The Legend of Zelda*, criada pela Nintendo. Como um bom fã, comprou todos os jogos que foram lançados até então e atualmente está jogando *Zelda: Twilight Princess* no seu *Wii* ao invés de terminar sua dissertação de Mestrado.

Após jogar por dezenas de horas, Felipe ainda não conseguiu passar de um mini game e conquistar o último *heart piece* que resta. O mini game consiste no seguinte: em um passeio de dragão, Link, o herói do jogo, deve estourar balões e acumular o máximo de pontos possíveis. Existem 3 tipos de balões, possivelmente com valores distintos. Ao estourar o i-ésimo balão, que é do tipo k, a pontuação total aumenta em V_k caso o último balão estourado seja de um tipo diferente do balão i, ou aumenta em o dobro do valor computado no último balão caso ele seja do mesmo tipo que o balão i (isso é considerado um bônus). Porém, esse bônus só pode dobrar de valor no máximo nove vezes, ou seja, o valor máximo que um balão do tipo k pode alcançar é *min(dobro do último balão,* $V_k \times 2^9$*)*. Se um balão não for estourado, ele não é contabilizado na soma final. V_k é o valor inicial de todos os balões do tipo k ($k \in \{a, b, c\}$), representado pelos valores A, B e C da entrada.

Os balões devem ser estourados na ordem em que aparecem, mas é possível decidir entre estourar ou não cada um dos balões para assim tentar aproveitar o bônus acumulado e aumentar a pontuação final. Ajude Felipe a calcular qual a maior pontuação possível de ser obtida no mini game para que ele possa tentar melhorar sua estratégia.

Entrada

A primeira linha da entrada contém um inteiro T, o número de casos de teste, seguido de $T \times 2$ linhas. Cada caso de teste é composto por duas linhas.

Na primeira linha de cada teste são dados três inteiros A, B e C ($1 \leq A, B, C \leq 10$), os valores iniciais dos três tipos de balões. Na segunda linha do teste é dada uma string S com no mínimo 1 e no máximo 100 caracteres e contendo apenas as letras a, b e c, os tipos dos balões que aparecerão. Inicialmente, balões do tipo a valem A, do tipo b valem B e do tipo c valem C.

Saída

Para cada caso de teste, imprima uma linha contendo a pontuação máxima possível de ser alcançada no jogo, considerando que Felipe é capaz de estourar todos os balões mas respeitando a ordem dada em S.

Exemplo de entrada	**Exemplo de saída**
2 1 2 4 acb 2 5 3 acaa	7 14

Observações sobre o exemplo de entrada

No primeiro caso, o melhor a fazer é estourar todos os balões.

No segundo caso, o ideal é não estourar o segundo balão, de forma que o primeiro balão valerá 2, o terceiro valerá 4 e o quarto valerá 8 devido ao bônus acumulado. Se todos os balões fossem estourados, o valor total seria 2+3+2+4=11.

D: Dentista

`File: dentista.[c|cpp|java|pas]`

Os dentistas são extremamente meticulosos em seu trabalho, tendo que agir com muita precisão em todas as suas atividades. Pedro é um dentista meticuloso como todos os outros. Infelizmente sua secretária não é muito organizada e, com o intuito de ajudar sempre os pacientes, aceita que eles marquem consultas no horário que quiserem, sem se preocupar com os demais horários marcados, ocasionando vários conítos de horários que muito incomodaram Pedro e os pacientes. Por exemplo, se uma consulta começar às 9 horas e durar 2 horas, nenhuma outra consulta deveria ser marcada para iniciar as 10 horas.

Ao perceber que sua agenda tinha coníito de horários, Pedro pediu sua ajuda para descobrir a maior quantidade de consultas que podem ser atendidas sem sobreposição.

Você deve escrever um programa que, dados os horários de início e término das consultas agendadas pela secretária, responda a quantidade máxima de consultas que podem ser atenditas sem sobreposição.

Entrada

A primeira linha de cada entrada contém um inteiro N ($1 \leq N \leq 10.000$) que indica quantas consultas a secretária marcou. Cada uma das N linhas seguintes contém um par de inteiros X e Y separados por um espaço em branco ($0 \leq X < Y \leq 1.000.000$) que representam, respectivamente, o horário de início e de término da consulta. Considere que se uma consulta inicia no exato instante em que outra termina não há sobreposição. Os horários de início são fornecidos em ordem, podendo haver mais de uma consulta que inicie no mesmo horário.

O final da entrada é indicado por uma linha que contém apenas um zero.

Saída

Seu programa deve imprimir uma única linha, contendo um inteiro que representa a quantidade máxima de consultas que podem ser atendidas sem que haja qualquer sobreposição.

Exemplo de entrada	**Exemplo de saída**
3 10 100 40 130 120 200 4 10 20 20 30 30 40 40 50 5 10 30 20 40 30 60 40 80 60 100 0	2 4 3

E: Marciano

File: marciano.[c|cpp|java|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.

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.

O final da entrada é indicado por uma linha que contém apenas quatro zeros, separados por espaços em branco.

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.

Exemplo de entrada	**Exemplo de saída**
10 20 30 30 10 10 10 7 2 4 4 3 0 0 0 0	S N S