



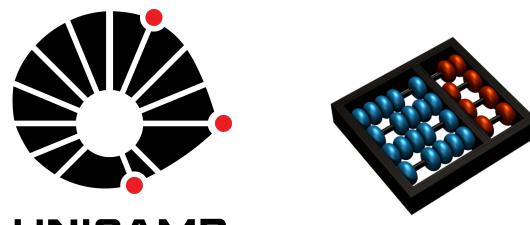
# Maratona de Programação dos Calouros

## Unicamp – 2019

18 de maio de 2019

A PROVA TEM DURAÇÃO DE 4:00 HORAS

Apoio:



Patrocínio:



# Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 20 páginas (não contando a folha de rosto), numeradas de 1 a 20. Verifique se o caderno está completo.
- 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.
- 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.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão e escritos na saída padrão. Utilize as funções padrão para entrada e saída de dados:
  - 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*
- 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.

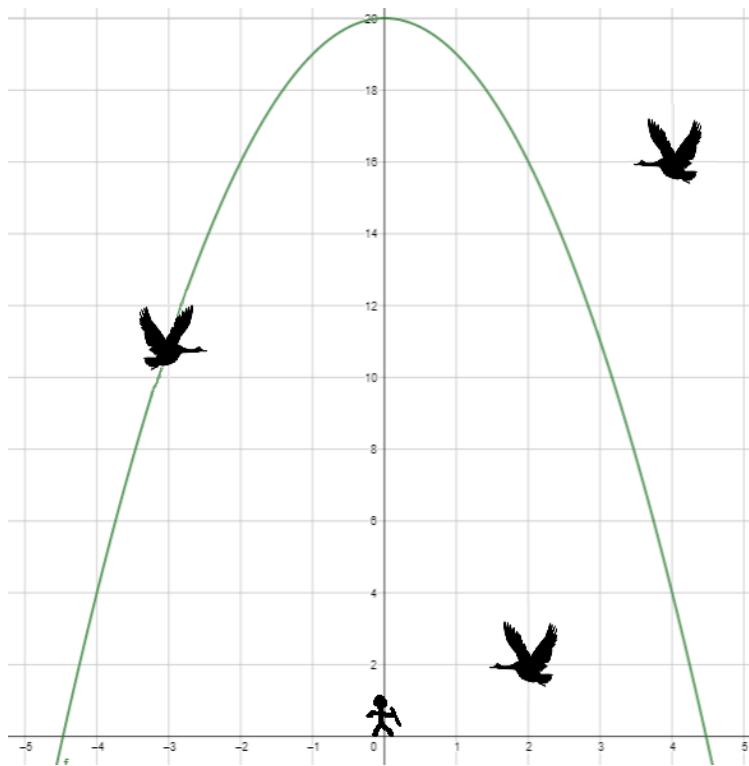
## A – Atirando em Gansos

Enquanto Erick levava o coffe-break da maratona dos calouros até o IC, o cheiro atraiu um bando de gansos selvagens que atacaram nosso amigo! Por sorte, Erick carregava consigo sua fiel zarabatana, e se preparou para abater as aves famintas.

Lembrando-se de suas aulas de Física I, ele concluiu que pode acertar qualquer ganso que tenha coordenadas  $(x, y)$  em relação a ele se e somente se  $y \leq A - Bx^2$ , onde  $A$  e  $B$  são duas constantes inteiras positivas.

Dadas as constantes  $A$  e  $B$ , e a lista das posições dos gansos, determine quantos deles podem ser abatidos.

No exemplo da figura abaixo, os gansos nas posições  $(2, 2)$  e  $(-3, 11)$  são abatidos por Erick, enquanto o ganso na posição  $(4, 16)$  escapa (a curva representa o limite de sua região de alcance).



### Entrada

A primeira linha contém três inteiros,  $A$ ,  $B$  e  $N$  respectivamente.  $A$  e  $B$  são as constantes descritas acima e  $N$  é o número de gansos.

Seguem  $N$  linhas, onde a  $i$ -ésima linha possui dois inteiros  $x_i$  e  $y_i$ , representando as coordenadas do  $i$ -ésimo ganso (em relação ao Erick, que está sempre na posição  $(0, 0)$ ).

### Saída

Imprima uma linha com um único número inteiro, o número de gansos abatidos.

### Restrições

- $1 \leq A \leq 100$

- $1 \leq B \leq 100$
- $1 \leq N \leq 100$
- $-100 \leq x_i \leq 100$
- $1 \leq y_i \leq 100$

## Exemplos

Entrada	Saída
20 1 3 2 2 -3 11 4 16	2

## B – Bolando Times

Um dos pontos mais debatidos na organização da maratona dos calouros foi o tamanho ideal para as equipes. Por um lado, se os times fossem muito pequenos, os alunos não fariam muitos amigos no evento. Por outro, se os times fossem grandes demais, os alunos poderiam brigar pelo tempo no computador.

A conclusão foi que o número de pessoas por equipe não importa realmente, desde que seja um número primo. É justamente isso que você deve fazer neste problema: dado um tamanho de equipe, descubra se ele corresponde a um número primo. Só lembrando que um número primo é um número natural maior que 1 que não pode ser obtido pelo produto de dois ou mais números naturais maiores que 1.

### Entrada

A entrada consiste de um único inteiro,  $N$ , o número que você deve verificar se é primo.

### Saída

Imprima uma única linha com um caractere ‘S’ se  $N$  for primo, e ‘N’ caso contrário.

### Restrições

- $1 \leq N \leq 10^{12}$

### Exemplos

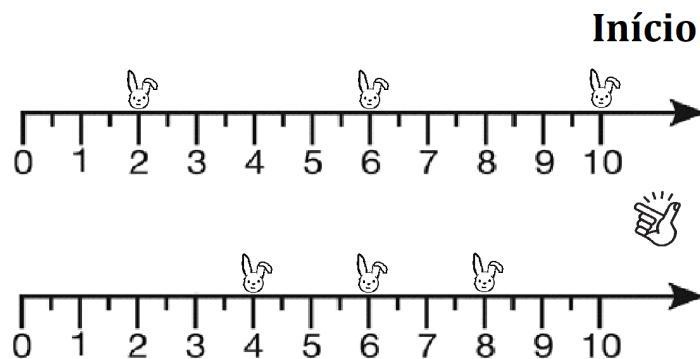
Entrada	Saída
7	S
99999999353	S
410338673	N

## C – Coelhos

Tiago decidiu abrir a premiação da maratona dos calouros com seu famoso número de mágica, o truque dos coelhinhos. O truque funciona da seguinte forma:

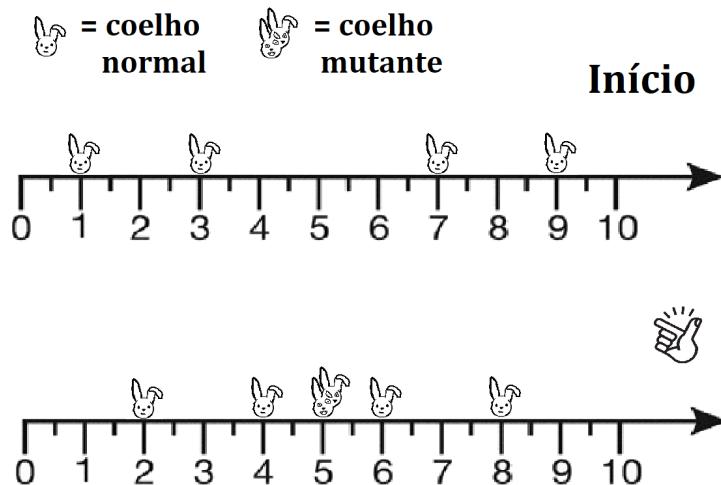
- Ele coloca  $N$  coelhos em uma linha reta, em diferentes posições.
- Ele começa a estalar os dedos.
- A cada estalo, os coelhos que estão na linha desaparecem.
- Logo em seguida, aparecem novos coelhos da seguinte forma: Para cada par de coelhos  $A$  e  $B$  que existia no grupo anterior, aparece um coelho no ponto médio das posições de  $A$  e  $B$ .

Veja, na figura abaixo, um exemplo onde ele estala os dedos uma vez:



Os coelhos estão inicialmente nas posições 2, 6 e 10. Quando Tiago estala os dedos, esses coelhos desaparecem, e aparecem coelhos nas posições 4 (por causa do par 2 e 6), 6 (por causa do par 2 e 10) e 8 (por causa do par 6 e 10).

Porém, Tiago sabe que essa mágica possui um risco: Se ele estalar os dedos, e dois ou mais coelhos aparecerem no mesmo lugar ao mesmo tempo, eles vão acabar se fundindo em uma aberração mutante saída diretamente de um conto de Lovecraft. Veja o exemplo abaixo:



Na imagem, inicialmente haviam coelhos nas posições 1, 3, 7 e 9. Depois que Tiago estalou os dedos, aparecem dois coelhos na posição 5, um por conta do par 3 e 7, e outro por conta do par 1 e 9 . A fusão dos dois gera um coelho mutante.

Para não traumatizar a plateia em um evento tão alegre, Tiago quer saber, dada uma configuração inicial, depois de quantos estalos apareceria o primeiro coelho mutante.

## Entrada

A primeira linha contém um inteiro  $N$ , o número inicial de coelhos.

Cada uma das  $N$  linhas seguintes contém um inteiro  $x_i$ , a posição do  $i$ -ésimo coelho.

## Saída

Imprima uma linha com um único inteiro, o número de estalos de dedo entre a configuração inicial e a primeira configuração que contenha pelo menos um coelho mutante. Se nunca surgir um coelho mutante, imprima apenas uma linha com  $-1$ .

## Restrições

- $1 \leq N \leq 5$
- $-10^9 \leq x_i \leq 10^9$
- Todos os  $x_i$  são distintos.

## Exemplos

Entrada	Saída
3 2 6 10	-1

Entrada	Saída
4 1 3 7 9	1

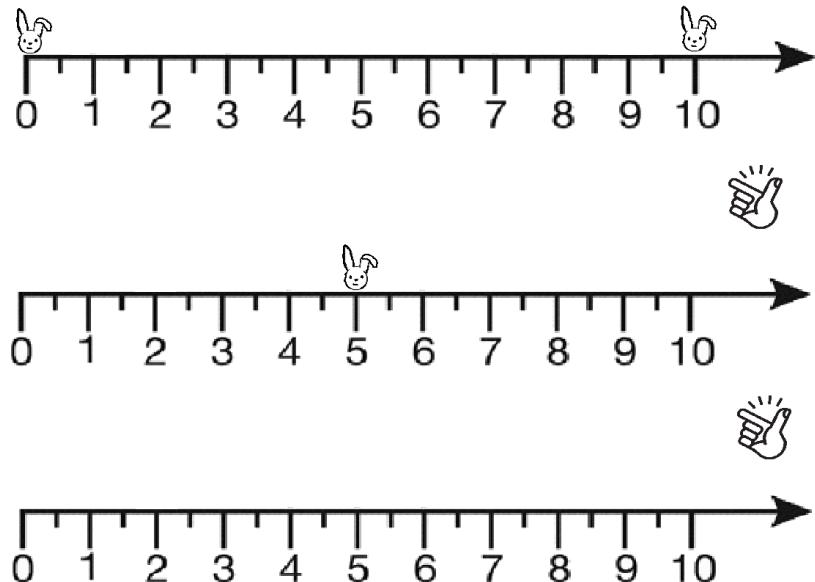
Entrada	Saída
2 1 10	-1

Entrada	Saída
4 2 4 6 10	2

## Explicação dos Exemplos

Os dois primeiros exemplos são explicados no enunciado. O terceiro caso é mostrado na imagem a seguir:

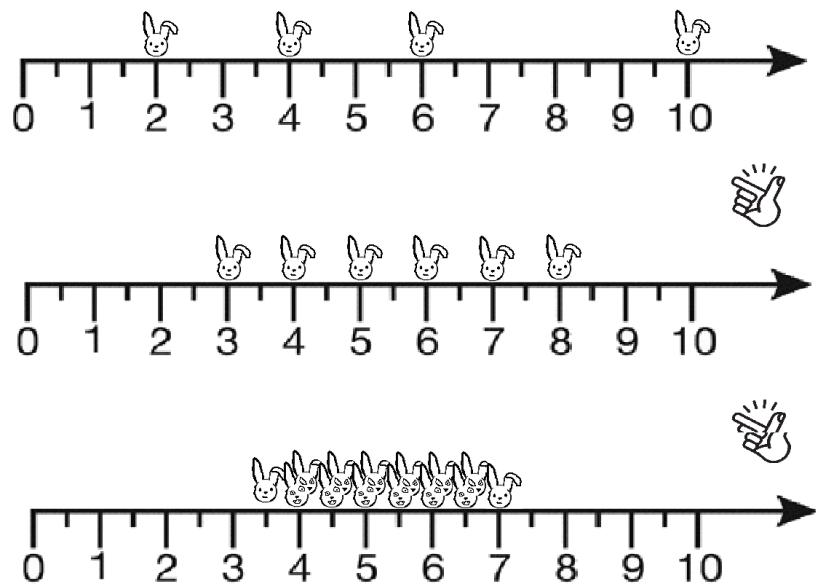
### Início



Como, após o segundo estalar de dedos, nenhum coelho aparece, nunca haverá um coelho mutante.

O quarto caso corresponde à imagem a seguir (repare que a partir do primeiro estalo podem aparecer coelhos em posições não inteiras):

### Início



# D – Designando Lugares

Chegou a hora de começar a competição, mas várias pessoas estão sentadas nos laboratórios do IC e pode não ser possível encontrar lugar para todos os times. Vamos considerar que um laboratório do IC tem  $N$  fileiras de cadeiras, cada uma com  $M$  colunas. As cadeiras podem estar livres ou ocupadas e você deve descobrir quantos times de três pessoas podem ser acomodados no laboratório descrito, dadas as seguintes condições:

- Os três integrantes do time devem sentar em cadeiras adjacentes da mesma fileira.
- As pessoas que já estão ocupando cadeiras do laboratório não podem ser movidas de lugar.

## Entrada

A primeira linha contém um inteiro  $N$ , o número de fileiras de cadeiras no laboratório. A segunda linha contém um inteiro  $M$ , o número de colunas de cadeiras no laboratório.

As  $N$  linhas seguintes contém  $M$  caracteres cada uma. Se o  $j$ -ésimo caractere da  $i$ -ésima linha for um ‘.’, então a  $j$ -ésima cadeira da  $i$ -ésima fileira do laboratório está livre. Caso contrário, o  $j$ -ésimo caractere da  $i$ -ésima linha será um ‘#’.

## Saída

O seu programa deve imprimir uma linha contendo um único inteiro, representando a quantidade de times que podem ser acomodados na sala.

## Restrições

- $1 \leq N \leq 50$
- $1 \leq M \leq 50$

## Exemplos

Entrada	Saída
5 10 .....#... ##### .#.#.#.... #.....# .....	9

Entrada	Saída
5 5 ..#.. .#..# #.#.。 ##..# .#.#	0

## E – Equipe Campeã

O MaratonIC deseja fazer um programa para determinar a equipe campeã da maratona dos calouros. A equipe campeã é aquela que resolve a maior quantidade de problemas. Empates no número de problemas resolvidos são classificados pelo tempo corrigido. Nesse caso, ganha a equipe que tem o menor tempo corrigido. O tempo corrigido da equipe é dado pela soma dos tempos corrigidos somente dos problemas corretamente resolvidos pela equipe. O tempo corrigido de um problema é dado pelo número de minutos decorridos desde o início da competição até o momento da primeira submissão correta somado com uma penalidade de 20 minutos por submissão incorreta feita anteriormente neste problema. Se persistir o empate, o MaratonIC fará um sorteio entre as equipes envolvidas.

O MaratonIC pede sua ajuda para elaborar um programa que determina o número de problemas resolvidos e o tempo corrigido da equipe campeã e se haverá necessidade de realizar um sorteio.

### Entrada

A primeira linha da entrada consiste em dois inteiros,  $E$  e  $P$ , correspondentes ao número de equipes e número de problemas no placar, respectivamente. A segunda linha contém um inteiro  $S$  correspondente ao número total de submissões realizadas pelas equipes. As  $S$  linhas seguintes representam as submissões. Cada linha  $i$  é composta por quatro inteiros  $t_i$ ,  $e_i$ ,  $p_i$  e  $v_i$ , tais que:

- $t_i$  representa o tempo transcorrido, em minutos, para a submissão  $i$ ;
- $e_i$  representa o índice da equipe que realizou a submissão  $i$ ;
- $p_i$  representa o índice do problema da submissão  $i$ ;
- $v_i$  representa se a submissão  $i$  obteve veredito de correta ( $v_i = 1$ ) ou incorreta ( $v_i = 0$ ).

### Saída

Caso a equipe campeã tenha que ser definida por sorteio, imprima uma linha com o texto “SORTEIO”. Caso contrário, imprima uma linha com três inteiros separados por espaços, representando: <índice da equipe campeã> <número de problemas resolvidos pela equipe campeã> <tempo corrigido da equipe campeã>.

### Restrições

- $1 \leq E \leq 10$ .
- $1 \leq P \leq 10$ .
- $1 \leq S \leq 1000$ .
- A lista de submissões é dada em ordem não-decrescente de tempo.
- Na lista de submissões não existem duas submissões da mesma equipe no mesmo tempo  $t_i$ .
- $1 \leq t_i \leq 240$ .
- $1 \leq e_i \leq E$ .
- $1 \leq p_i \leq P$ .
- $0 \leq v_i \leq 1$ .

## Exemplos

Entrada	Saída
2 2 5 10 1 2 0 20 2 1 0 30 1 2 1 40 2 1 1 50 1 1 1	1 2 100

Entrada	Saída
3 5 5 10 3 2 1 22 2 5 0 40 2 5 1 55 1 1 0 70 3 2 1	3 1 10

Entrada	Saída
2 4 8 10 1 1 1 10 2 4 1 20 1 2 1 20 2 3 1 30 1 3 1 30 2 2 1 40 1 4 1 40 2 1 1	SORTEIO

# F – Festa Universitária

Joãozinho, aluno de universidade pública, foi a uma festa beber. Após tamanha balbúrdia, Joãozinho corre sério perigo: seu campo de visão está totalmente distorcido e ele corre o risco de não chegar em casa.

O campo de visão de Joãozinho pode ser reduzido a uma matriz de inteiros  $N \times N$ . Na festa, ele tomou  $K$  copos de bebida e cada um rotacionou sua vista em  $90^\circ$  no sentido anti-horário. Ajude Joãozinho a chegar em casa reconstruindo a matriz de visão original dele (ou seja, desfaça todas as rotações provocadas pelo álcool).

## Entrada

As duas primeiras linhas têm números inteiros representando  $N$  e  $K$ , o tamanho da matriz e o número de copos de bebida que ele tomou, respectivamente.

As  $N$  linhas seguintes possuem  $N$  inteiros  $a_{ij}$  cada, separados por espaços, representando o campo de visão atual de Joãozinho (após as rotações causadas pelo álcool).

## Saída

Imprima  $N$  linhas com  $N$  inteiros  $a_{ij}$ , separados por espaços, representando o campo de visão original.

**Atenção:** não coloque um espaço depois do último número de cada linha.

## Restrições

- $1 \leq N \leq 100$
- $1 \leq K \leq 10^9$
- $0 \leq a_{ij} \leq 9$

## Exemplos

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

Entrada	Saída
5 42 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4	4 3 2 1 0 4 3 2 1 0 4 3 2 1 0 4 3 2 1 0 4 3 2 1 0

## G – Guto, Entregador de Pizzas

Guto é dono de uma pizzaria. Ele recebeu um grande pedido de pizzas para entregar na Maratona dos Calouros da Unicamp.

Guto além de ser o dono da pizzaria é também o entregador. Ele tem um carro que irá usar para levar as pizzas da pizzaria até o IC. A distância dessa viagem é igual a  $D$  quilômetros.

Contudo, o carro de Guto não está em boas condições. O carro sempre quebra a cada  $K$  quilômetros percorridos e Guto precisa gastar  $T$  segundos consertando-o (ele também é mecânico, uau). Uma vez consertado, o carro poderá andar novos  $K$  quilômetros até quebrar novamente.

Para percorrer um quilômetro de carro Guto gasta  $A$  segundos. Contudo, Guto é um maratonista e pode também realizar trajetos correndo! Para percorrer um quilômetro correndo Guto gasta  $B$  segundos.

Por isso, sua tarefa é encontrar o tempo mínimo que Guto precisa gastar para entregar suas pizzas. Para isso, considere que a qualquer momento Guto pode abandonar o seu carro e prosseguir sua entrega correndo.

### Entrada

A entrada tem 5 linhas com um inteiro cada representando, respectivamente, os valores  $D$ ,  $K$ ,  $A$ ,  $B$ ,  $T$ , onde:

- $D$  é a distância em quilômetros entre a pizzaria e o destino (IC).
- $K$  é a distância em quilômetros que o carro consegue percorrer antes de quebrar.
- $A$  é o tempo gasto para que Guto percorra um quilômetro de carro.
- $B$  é o tempo gasto para que Guto percorra um quilômetro correndo.
- $T$  é o tempo gasto para que Guto conserte o carro quebrado.

### Saída

Imprima em uma linha o tempo mínimo que Guto precisa gastar para percorrer a distância  $D$  da origem ao destino.

### Restrições

- $1 \leq D \leq 10^{12}$ .
- $1 \leq K, A, B, T \leq 10^6$ .
- $A < B$ .

### Exemplos

Entrada	Saída
5 2 1 4 10	14

Entrada	Saída
5 2 1 4 5	13

# H – Halteres

Pensando em se preparar para carregar o cilindro de hélio pela escadaria do IC no dia da maratona, digoBoy resolveu ir à academia se exercitar.

Chegando lá, ele percebeu que havia alguma coisa errada: os halteres estavam mal distribuídos entre os suportes. Para digoBoy, cada um dos suportes deveria ter exatamente  $P$  halteres, de pesos 1, 2, …  $P$ . Ele resolveu reorganizar a academia mudando alguns halteres de suporte, mas antes quer ter certeza que é possível criar a distribuição descrita.

Observação: halteres são aqueles pesos de exercício feitos para serem levantados com uma mão.

## Entrada

A primeira linha da entrada consiste em dois inteiros,  $S$  e  $P$ , que representam respectivamente o número de suportes e o valor  $P$  descrito no enunciado.

As  $S$  linhas seguintes descrevem cada um dos suportes. A  $i$ -ésima começa com um inteiro  $N_i$ , o número inicial de halteres no  $i$ -ésimo suporte. Seguem  $N_i$  inteiros positivos  $w_{ij}$  (para  $1 \leq j \leq N_i$ ), representando os pesos dos halteres que estão inicialmente no  $i$ -ésimo suporte.

## Saída

Imprima uma linha com “S” caso a organização descrita seja possível, e “N” caso contrário.

## Restrições

- $1 \leq S \leq 30$
- $1 \leq P \leq 30$
- $1 \leq N_i \leq 30$
- $1 \leq w_{ij} \leq 30$

## Exemplos

Entrada	Saída
2 3 3 1 3 3 3 1 2 2	S

Entrada	Saída
2 4 4 1 2 3 3 4 1 2 2 4	N

Entrada	Saída
2 3 3 3 1 3 3 1 2 2	S

Entrada	Saída
2 2 2 1 2 2 1 2	S

Entrada	Saída
2 3 3 1 2 3 4 1 2 3 4	N

Entrada	Saída
2 3 2 1 2 4 1 3 2 3	S

# I – Yan

Tiago é um aluno muito dedicado que vai a todas as aulas de sua faculdade, mas ele é um aluno muito inquieto e que perturba os colegas se fica sem problemas interessantes para pensar. Seu amigo Yan, que também vai a todas as aulas com ele, criou um problema para ajudá-lo.

Tiago precisa contar quantas vezes a subsequência “yan” aparece em uma string. Subsequências não precisam ser partes contínuas do texto. Por exemplo, em “ytiagon” a subsequência “yan” aparece 1 vez.

Tente resolver o problema você também!

## Entrada

Uma única linha com uma string  $S$ . Todos os caracteres da string são letras minúsculas do alfabeto, sem espaços.

## Saída

Imprima em uma única linha a quantidade de vezes em que a subsequência “yan” aparece em  $S$ .

## Restrições

- A string tem no máximo 5000 caracteres.

## Exemplos

Entrada	Saída
yingyangyo	2
yyann	4
yayananya	8

## Explicação dos Exemplos

Vamos enumerar as posições da string de entrada a partir de zero. Assim, no primeiro exemplo temos duas subsequências “yan”, uma formada por pelas posições (0, 5, 6) e outra formada pelas posições (4, 5, 6).

No segundo exemplo temos quatro subsequências “yan”: (0, 2, 3), (1, 2, 3), (0, 2, 4) e (1, 2, 4).

## J – Juiz de Vôlei

A quantidade de efeito que os jogadores de vôlei estão conseguindo colocar na bola é realmente assustadora! Devido às incríveis distâncias e velocidades envolvidas, os juízes humanos já não conseguem acompanhar a trajetória da bola. Por isso, a Faculdade de Educação Física (FEF) decidiu contratar você, aluno do IC, para desenvolver um juiz eletrônico para os jogos.

Neste momento, você quer detectar se um saque de um lado para o outro da quadra passou por cima da rede ou por fora da rede (já que, pelas regras, não é permitido passar por fora).

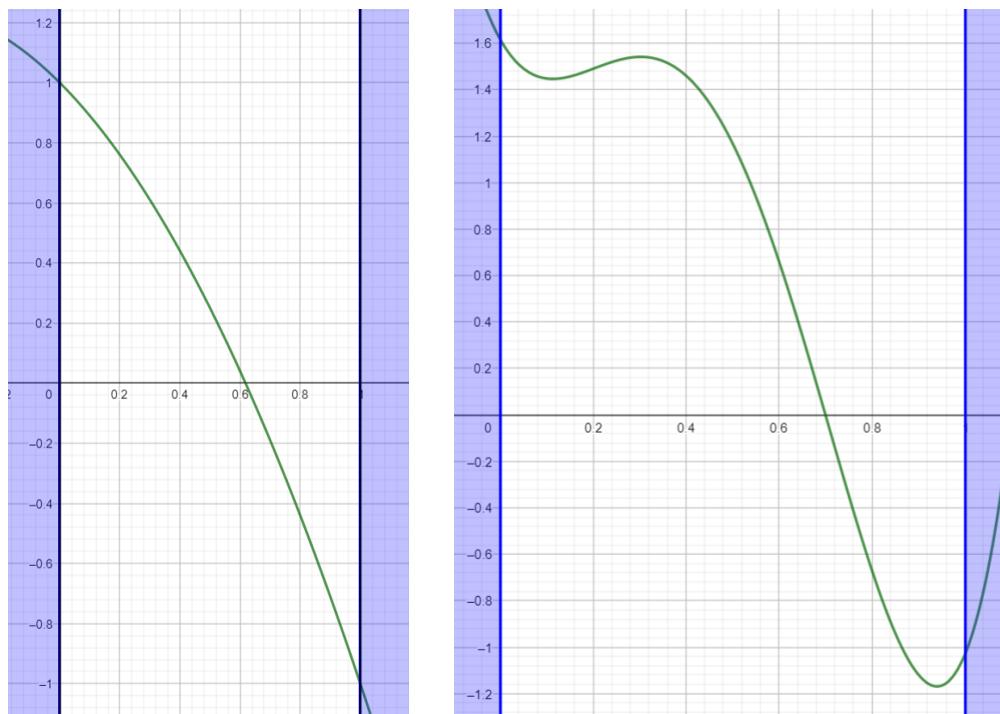
O seu juiz eletrônico acompanha o jogo usando uma câmera posicionada em cima do campo. No seu sistema, o campo é modelado por um plano cartesiano de posições  $(x, y)$ , para os valores de  $x$  entre 0 e 1 (limites laterais da quadra). A rede fica sobre a reta  $y = 0$ , a parte do plano com  $y > 0$  é o campo de um time e a parte com  $y < 0$  é o campo do outro time.

Pelos dados já coletados, você descobriu que a trajetória de qualquer saque pode ser descrita por um polinômio de grau 5.

O sistema que você está construindo já consegue analisar as imagens e descobrir qual é o polinômio que descreve a trajetória da bola. Seja  $f(x) = c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$  a função polinomial da trajetória. A bola parte da posição  $(0, f(0))$  no campo superior ( $f(0) > 0$ ) e vai até a posição  $(1, f(1))$  no campo inferior ( $f(1) < 0$ ), pela trajetória de  $f(x)$ . Você também sabe que a bola cruza a reta  $y = 0$  (onde fica a rede) somente uma vez para  $0 \leq x \leq 1$ .

Seu objetivo é encontrar o valor de  $x$  no momento que a bola cruza a reta  $y = 0$  para saber se ela passou por cima ou por fora da rede. Ou seja, você quer encontrar  $x$  tal que  $f(x) = 0$ .

As duas figuras abaixo ilustram a trajetória da bola nos exemplos de entrada. A região válida da quadra também está demarcada nas figuras ( $0 \leq x \leq 1$ ). Na figura da direita, por exemplo, podemos ver que a bola cruza a rede quando  $x = 0.7$ .



## Entrada

A entrada tem 6 linhas com um número real cada representando, respectivamente, os valores  $c_0, c_1, c_2, c_3, c_4$  e  $c_5$  descritos no enunciado.

Para ler um número da entrada, você pode usar o comando `c0 = float(input())` em Python 3, por exemplo.

## Saída

O seu programa deve imprimir uma linha contendo um único número real com 9 casas decimais, o valor de  $x$  tal que  $f(x) = 0$  e  $0 \leq x \leq 1$ .

Tenha cuidado com a precisão necessária, porque uma resposta de 0.699999999 será considerada errada se o correto for 0.700000000. Para imprimir a saída, você pode usar o comando `print("%.9f" % (x))` em Python 3.

## Restrições

- $-1000 \leq c_i \leq 1000$ , para  $0 \leq i \leq 5$ .
- $f(0) > 0$  e  $f(1) < 0$ , onde  $f(x) = c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$ .
- Existe um único valor de  $x$  tal que  $0 \leq x \leq 1$  e  $f(x) = 0$ .
- O valor real  $x$  da resposta será tal que  $x + 10^{-10}$  e  $x - 10^{-10}$  ainda arredondam para o mesmo valor com 9 casas decimais. Por exemplo, a resposta para o primeiro exemplo com maior precisão seria 0.61803398875. Porém, tanto 0.61803398885 quanto 0.61803398865 arredondam para 0.618033989 com 9 casas decimais.

## Exemplos

Entrada	Saída
1.0 -1.0 -1.0 0 0 0	0.618033989

Entrada	Saída
1.617 -3.549 23.12 -47.02 20.8 4	0.700000000

## K – Kururu Pulex

Na Unicamp podemos encontrar vários animais ao andar de um instituto a outro ou para ir ao bandejão, como gambás, capivaras (nossa favorita) e sapos (um animal que se move através de pulos).

No mesmo laboratório em que está o ET de Varginha, certos alunos querem estudar uma espécie de sapo, *Kururu pulex*, que possui apenas dois tipos de pulos, um que percorre uma distância inteira  $A$  e outro que percorre uma distância inteira  $B$ .

Os alunos colocaram o sapo em um tubo infinito (sim, temos isso, é o laboratório do ET de Varginha), na marca de posição 0, de modo que o sapo pode se mover em ambos os sentidos dentro do tubo através de pulos. Eles desejam saber se o sapo consegue atingir o ponto na marca de posição  $D$  do tubo, você pode ajudá-los com isso?

### Entrada

A primeira linha contém um inteiro  $T$ , o número de casos de teste.

Cada linha seguinte contém três inteiros separados por espaço  $A$ ,  $B$ ,  $D$ , as distâncias dos pulos que o sapo consegue fazer e o ponto que querem saber se o sapo consegue atingir.

Para ler 3 inteiros em uma linha em Python 3, pode-se usar o comando `A, B, D = map(int, input().split())`.

### Saída

Imprima “S” em uma linha caso o sapo consiga atingir a posição  $D$ , imprima “N” em uma linha caso contrário.

### Restrições

- $1 \leq T \leq 100$
- $1 \leq A, B \leq 10^5$
- $-10^5 \leq D \leq 10^5$

### Exemplos

Entrada	Saída
4	S
5 2 1	S
3 20 -6	S
2 10 0	N
2 4 1	

## L – Língua do P

Arthur está passando por uma fase rebelde, e decidiu que agora só vai se comunicar usando a língua do P. Para quem não sabe, uma frase em português pode ser traduzida diretamente para a língua do P adicionando-se a letra ‘p’ minúscula antes de cada letra da sentença original.

Por exemplo, a frase “**eu gosto de churros**” em português é traduzida para “**pepu pgpopspptpo pdpe pcphpuprprprops**” na língua do P.

Os outros membros da organização da maratona dos calouros estão com dificuldade para entender as mensagens de Arthur, e pediram a sua ajuda! Faça um programa que receba uma frase na língua do P e imprima a frase em português correspondente.

### Entrada

A entrada consiste de uma única linha que contém uma frase válida na língua do P. Os únicos caracteres que aparecerão na frase serão espaços e letras minúsculas de ‘a’ a ‘z’.

### Saída

Imprima uma linha com a frase traduzida para o português.

### Restrições

- A entrada é uma frase válida na língua do P.
- Não há espaços consecutivos na frase.
- A frase dada não começa nem termina com espaços.
- O número máximo de caracteres da string de entrada é 1000.

### Exemplos

Entrada	Saída
pppapppa pdpo pppapppa	papa do papa
pepl pppspy pkpopnpgprpo	el psy kongro
pppppp	ppp