

A (Au | Mal)dição de Abraão

Abraão é conhecido entre os colegas por ter uma audição muito boa. Os amigos de Abraão estão sempre criando novos desafios com os mais variados tipos de sons, mas Abraão sempre consegue superá-los, o que deixa todos muito surpresos e admirados.

O último desafio criado envolve um pouco de matemática, que, infelizmente, não é o forte de Abraão. O desafio é o seguinte, Abraão é vendado e levado a uma sala com n pessoas (o valor de n não é conhecido nesse momento). Cada pessoa segura uma taça e todas elas estão prestes a fazer um brinde. Após o brinde, pergunta-se a Abraão: "qual o valor de n "? Abraão está confiante que consegue contar todas as colisões entre taças. No entanto, ele não faz ideia de como obter o valor de n a partir do número de colisões contadas.

É nesse ponto que suas habilidades de matemática e programação serão requisitadas! Dado o número de colisões identificadas por Abraão, faça um programa que retorne o valor de n . Você pode assumir que:

- Sempre haverá pelo menos uma pessoa participando do brinde;
- Cada uma das n taças colide com todas as demais;
- Cada par de taças colide uma única vez durante o brinde;
- Abraão é capaz de identificar corretamente todas as colisões entre taças, isto é, ele não conta nenhuma colisão a menos.
- Em alguns casos, pode ser que ele conte colisões a mais.

Entrada

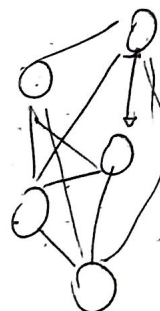
A entrada será feita pela entrada padrão. A primeira linha contém um inteiro T ($1 \leq T \leq 103$), representando o número de casos de teste (desafios feitos a Abraão). Cada uma das T seguintes linhas conterá um inteiro K ($0 \leq K \leq 106$). O valor de K representa o número de colisões identificadas por Abraão durante o desafio.

Saída

A saída deve ser feita na saída padrão. Para cada valor de K da entrada, seu programa deve imprimir na saída padrão o valor de n correspondente, i.e., o número de pessoas que participaram do brinde. Cada valor de n deve ser sucedido por um caractere de nova linha ('\n').

Exemplo

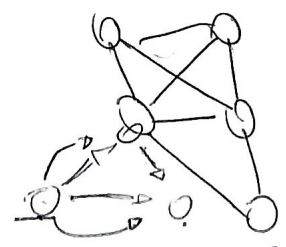
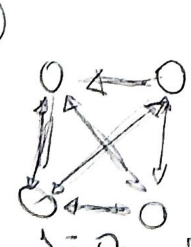
Entrada	Saída
3	1
0	2
1	3



$$a_3 = a_1(3-1)$$

$$a_3 = a_2(3-1)$$

$3! = 6$
 $4! = 24$
 $5! = 120$
 $6! = 720$
 $7! = 5040$
 $8! = 40320$
 $9! = 362880$
 $10! = 3628800$
 $11! = 39916800$
 $12! = 479001600$
 $13! = 6227020800$
 $14! = 87178291200$
 $15! = 1307674368000$
 $16! = 20922789888000$
 $17! = 355687420800000$
 $18! = 6402373705600000$
 $19! = 121645100800000000$
 $20! = 2432902008176640000$



$$3 \cdot 2 \cdot 1 = 6$$

Expressões lógicas

Dada uma expressão lógica (com variáveis Booleanas e operadores "and" e "not"), classificar a expressão em:

1. "trivialmente falsa", quando qualquer atribuição de valores para as variáveis booleanas resultar em uma avaliação falsa para a expressão, ou,
2. "nem trivialmente verdadeira, nem trivialmente falsa", quando diferentes atribuições de valores para as variáveis resultarem em diferentes avaliações para a expressão.

Entrada

A expressão será dada como uma conjunção. Variáveis serão denotadas com letras maiúsculas (de A a H). Um operador unário de negação "not" poderá ser usado antes de uma variável. O operador binário de conjunção "and" será usado entre as variáveis (possivelmente negadas). Um espaço em branco será usado como separador. A primeira linha da entrada indica o número de expressões lógicas, que serão dadas uma por linha.

Saída

A saída deverá imprimir a classificação da expressão de entrada.

Exemplo

Entrada	Saída
6	trivialmente falsa
not A and A	nem trivialmente verdadeira, nem
A	trivialmente falsa
A and B and C and D and E	nem trivialmente verdadeira, nem
not A and B and C	trivialmente falsa
not A and not B	nem trivialmente verdadeira, nem
A and not B and B and C	trivialmente falsa
	nem trivialmente verdadeira, nem
	trivialmente falsa
	trivialmente falsa