

Problema A

Corrida de lesmas

lesmas.c lesmas.cpp lesmas.java

A corrida de lesmas é um esporte que cresceu muito nos últimos anos, fazendo com que várias pessoas dediquem suas vidas tentando capturar lesmas velozes para treina-las e faturar milhões em corridas pelo mundo. Porém a tarefa de capturar lesmas velozes não é uma tarefa muito fácil, pois praticamente todas as lesmas são muito lentas. Cada lesma é classificada em um nível dependendo de sua velocidades em cm h^{-1} :

Nível 1: Se velocidade menor que 10 cm h^{-1} .

Nível 2: Se velocidade maior ou igual a 10 cm h^{-1} e menor que 20 cm h^{-1} .

Nível 3: Se velocidade maior ou igual a 20 cm h^{-1} .

Sua tarefa é identificar qual nível de velocidade da lesma mais veloz de um grupo de lesmas.

Entrada

A entrada consiste múltiplos casos de teste, e cada um consiste em duas linhas: A primeira linha contém um inteiro L representando o número de lesmas do grupo, e a segunda linha contém L inteiros v representando as velocidades de cada lesma do grupo. O final da entrada é indicado pelo fim do arquivo.

Limites

$$1 \leq L \leq 500$$

$$1 \leq v \leq 50$$

Saída

Para cada caso de teste imprima uma linha com um inteiro indicando o nível de velocidade da lesma mais veloz do grupo.

Exemplo de entrada / saída

| Entrada | Saída |
|-------------------------------|-------|
| 10 | 3 |
| 10 10 10 10 15 18 20 15 11 10 | 1 |
| 10 | 2 |
| 1 5 2 9 5 5 8 4 4 3 | |
| 10 | |
| 19 9 1 4 5 8 6 11 9 7 | |

Problema B

Ataque Alienígena

ataque.c ataque.cpp ataque.java

Uma aldeia distância chamada Zugai, está sendo alvo de vários ataques alienígena. Em cada ataque aparecem grandes triângulos desenhados na plantação. Os aldeões tentaram calcular a área afetada pelo ataque, porém só conseguiram medir os comprimentos dos lados dos triângulos. Então pediram sua ajuda para calcular a área afetada.

Sua tarefa é calcular a área do triângulo desenhado, sabendo a medida dos três lados.

Entrada

A entrada consiste de múltiplos ataques. A primeira linha de cada ataque contém um inteiro N que indica o número de triângulos desenhados durante o ataque, e em cada uma das próximas N linhas haverá três inteiros x, y, z representando as medidas dos lados de triângulo desenhado. A entrada termina quando N igual a 0.

Limites

$$1 \leq N \leq 500$$

$$1 \leq x, y, z \leq 300$$

Saída

Imprima na primeira linha de cada ataque "Ataque #ID" (sem aspas), com o ID representando o identificador do ataque, começando em ID igual a 1, e para cada triângulo desenhado imprima uma linha com a área do triângulo arredondada para 4 casas decimais. Sempre deixe uma linha em branco entre as saídas de cada ataque (Não imprima uma linha após do último ataque). Para melhor compreensão, observe os exemplos de entrada e saída.

Exemplo de entrada / saída

| Entrada | Saída |
|---------|-----------|
| 2 | Ataque #1 |
| 3 4 5 | 6.0000 |
| 6 6 6 | 15.5885 |
| 2 | |
| 5 8 10 | Ataque #2 |
| 8 9 15 | 19.8100 |
| 3 | 29.9333 |
| 5 3 5 | |
| 5 10 12 | Ataque #3 |
| 2 2 2 | 7.1545 |
| 0 | 24.5446 |
| | 1.7321 |

Problema C

Sapo não lava o pé

sapo.c sapo.cpp sapo.java

Em uma grande lagoa, mora um sapo conhecido por não querer lavar seus pés. Neste momento ele está posicionado em uma pedra no centro da lagoa querendo pular para um dos troncos que estão boiando ao seu redor, porém além de não querer lavar seus pés, ele também é muito preguiçoso, então ele irá pular para o tronco mais próximo da pedra onde está posicionado.

Considere a lagoa como um plano cartesiano, que a pedra onde o sapo está inicialmente está na posição $(0, 0)$ do plano e que cada tronco é representado por uma linha (x_i, y_i, x_f, y_f) . Sua tarefa é informar qual tronco é mais próximo do sapo.

Entrada

A entrada consiste de múltiplas linhas. A primeira linha contém um inteiro N que indica o número de troncos próximos ao sapo e cada uma das próximas N linhas contém quatro inteiros x_i, y_i, x_f e y_f indicando as posições dos pontos das extremidades do tronco, (x_i, y_i) e (x_f, y_f) .

Limites

$$1 \leq |N| \leq 10000$$
$$-10000 \leq x, y \leq 10000$$

Saída

Imprima uma única linha contendo dois números separados por um espaço, o identificador do tronco mais próximo e a distância que o sapo terá que pular arredondada para quatro casas decimais. Por exemplo, se o tronco mais próximo for o 4º tronco fornecido, imprima 4 e a distância até ele. Caso ocorra empates, o tronco escolhido será o com menor identificador.

Exemplo de entrada / saída

Entrada

```
6
0 2 2 2
3 1 5 1
-2 0 -1 3
0 4 2 6
2 4 5 4
-5 5 -2 5
```

Saída

```
3 1.8974
```

Problema D

Matriz de Potência

matriz.c matriz.cpp matriz.java

Uma matriz é uma Matriz de Potências se atende 3 pré-requisitos:

1. É uma matriz quadrada.
2. A primeira linha é formada apenas por 1's.
3. Para todo elemento $M[i][j]$ com $j! = 1$, $M[i][j] == M[i][2]^{j-1}$.

Por exemplo:

$$P = \begin{pmatrix} 1 & M[1][2]^1 & M[1][2]^2 & \dots & M[1][2]^{n-1} \\ 1 & M[2][2]^1 & M[2][2]^2 & \dots & M[2][2]^{n-1} \\ 1 & M[3][2]^1 & M[3][2]^2 & \dots & M[3][2]^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & M[n][2]^1 & M[n][2]^2 & \dots & M[n][2]^{n-1} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix}^0 == \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix}^1 == \begin{pmatrix} 1 & 3 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix}^2 == \begin{pmatrix} 4 & 9 & 3 \\ 4 & 9 & 3 \\ 4 & 9 & 3 \end{pmatrix}$$

Sua tarefa é descobrir se uma matriz quadrada pode ser transformada em uma Matriz de Potências utilizando dois tipos de operações:

1. Troca(x, y): Inverte as posições das colunas x e y da matriz.
2. Transposta(): A matriz é transposta.

Por exemplo:

$$P = \begin{pmatrix} 9 & 4 & 16 \\ 1 & 1 & 1 \\ 3 & 2 & 4 \end{pmatrix} \rightarrow Transposta(P) = \begin{pmatrix} 9 & 1 & 3 \\ 4 & 1 & 2 \\ 16 & 1 & 4 \end{pmatrix} \rightarrow Troca(1, 2) = \begin{pmatrix} 1 & 9 & 3 \\ 1 & 4 & 2 \\ 1 & 16 & 4 \end{pmatrix} \rightarrow$$

$$Troca(2, 3) = \begin{pmatrix} 1 & 3 & 9 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \end{pmatrix}$$

Logo P pode ser transformada em uma Matriz de Potência.

Entrada

A entrada consiste de múltiplas linhas. A primeira linha contém um inteiro C que indica o número de casos de teste. Em seguida, em cada caso de teste a primeira linha contém um inteiro N que indica o número de linhas e colunas da matriz, em seguida N linhas, cada uma com N inteiros d representando os elementos da matriz .

Limites

$$3 \leq N \leq 7$$
$$-50000 \leq d \leq 50000$$

Saída

Imprima em uma única linha para cada caso de teste a "Potencia" (sem aspas) caso a matriz possa ser transformada, ou "Nao Potencia" (sem aspas) caso contrário.

Exemplo de entrada / saída

| Entrada | Saída |
|---------|--------------|
| 3 | Potencia |
| 3 | Potencia |
| 16 1 4 | Nao Potencia |
| 1 1 1 | |
| 9 1 3 | |
| 3 | |
| 25 36 9 | |
| 1 1 1 | |
| 5 6 3 | |
| 3 | |
| 9 35 25 | |
| 3 6 5 | |
| 1 1 1 | |

Problema E

Sopa de letrinhas

sopa.c sopa.cpp sopa.java

Um restaurante é famoso por sua inigualável sopa, conhecida como Sopa Impressionante das Letras. Essa sopa é servida em um prato quadrado com dimensão de cada lado igual a $10cm$ e contendo 100 letrinhas, sendo que cada letrinha ocupa um espaço de $1cm^2$. Assim podemos ver um prato como uma matriz 10×10 e cada célula da matriz contendo uma letra.

Sabendo as letras são apenas as letras maiúsculas do alfabeto latino (sem as letras K, W e Y), sua tarefa é imprimir o número de ocorrências de determinada letra no prato.

Entrada

A entrada consiste de múltiplos casos de teste. Cada caso de teste consiste em onze linhas, primeira linha contém um caractere C que indica a letra que contaremos o número de ocorrências seguido de dez linhas, cada uma contendo dez letras, correspondente ao prato de letrinhas. O final da entrada é indicado pelo fim de arquivo.

Limites

$$A \leq C \leq X$$

Saída

Para cada caso de teste, imprima uma única linha contendo o número de ocorrências da letra C na Sopa Impressionante das Letras.

Exemplo de entrada / saída

| Entrada | Saída |
|------------|-------|
| A | 3 |
| IQRSQVSCUT | 6 |
| AXGUZCQBJP | |
| JLABATSDDI | |
| UVSNTJXFEX | |
| ETFZHLSMOU | |
| HGXMPUJHJC | |
| USBBZFISHS | |
| GIZUTTIHOP | |
| NNDEHMOQRI | |
| SJIROTMCEH | |
| G | |
| BDZJMPACDD | |
| GPUDICPBNA | |
| UBVGCXNJJT | |
| MNUVBXDNAZ | |
| RGEALUDXCR | |
| GGQTPZODQA | |
| QUXTHZFJQM | |
| NSXJXXFVZB | |
| FOOMZCBURQ | |
| AXVLQJBIIG | |

Problema F

Slimes

slimes.c slimes.cpp slimes.java

Slimes são criaturas mágicas, com o corpo feito de uma gosma ácida, e por isso muito perigosos para aventureiros descuidados. O ciclo de vida dos slimes é um tanto curioso, ele leva A semanas para chegar a fase adulta, e a cada semana posterior ele gera uma nova cria (que demorará A semanas para se tornar adulta) e depois de um determinado número de semanas V , ele se auto-destrói e renasce como uma cria. Perceba que no final da semana V o slime se auto-destrói e não se multiplica, apenas renasce. Sendo A e V determinados pela região onde o slime vive.

Um aventureiro ficou curioso com a quantidade de slimes que um slime poderia gerar em determinado tempo T . Sua tarefa é indicar quantos slimes estão vivos depois de uma quantidade T de semanas, considerando que inicialmente existe apenas **um** slime no **início** da vida dele.

Por exemplo, para $T = 6$, $A = 1$ e $V = 3$:

- Semana 1 - 2: Slime com idade de zero semanas fica com idade uma semana (Total = 1 slime).
- Semana 2 - 3: O slime com idade de uma semana gera um slime com idade de zero semanas e envelhece uma semana (Total = 2 slime).
- Semana 3 - 4: O slime com idade de zero semanas envelhece e o slime com idade de duas semanas se auto-destrói ao fim da 3ª semana e gera um slime de idade zero (Total = 2 slime).
- Semana 4 - 5: O slime de idade igual a zero envelhece e o slime de idade igual a um gera um novo slime de idade zero e envelhece (Total = 3 slime).
- Semana 5 - 6: O slime de idade igual a zero envelhece, o slime de idade igual a um gera um novo slime e envelhece, já o slime com idade igual a dois, se auto-destrói e gera um novo slime de idade zero (Total = 4 slime).

Entrada

A entrada consiste em múltiplos casos de teste, e cada um consiste em uma linha com três inteiros T , A e V , onde T é a quantidade de semanas decorridas, A a quantidade de semanas para um slime se tornar adulto e V a quantidade de semanas que um slime vive. O final da entrada é indicado pelo fim do arquivo.

Limites

$$1 \leq T \leq 80$$

$$1 \leq A \leq 5$$

$$1 \leq V \leq 20$$

$$A \leq V$$

Saída

Para cada caso de teste imprima uma linha com um inteiro indicando a quantidade de slimes vivos na região depois de T semanas.

Exemplo de entrada / saída

| Entrada | Saída |
|---------|----------|
| 6 1 3 | 4 |
| 20 4 10 | 71 |
| 80 5 6 | 1 |
| 50 2 20 | 82775023 |