

# Problema A

## Palabras encadenadas

Un juego común entre los niños de antes, cuando las consolas no eran portátiles y los móviles sólo servían para llamar, era el de palabras encadenadas. Este juego consiste en que dos jugadores van diciendo una palabra alternativamente, con la condición de que la primera sílaba de la palabra del segundo jugador sea igual a la última sílaba de la palabra del primero.

Nuestro trabajo consistirá en, dada una lista de  $N$  palabras, encontrar cuántas parejas de palabras encadenadas hay en total. Para simplificar un poco nuestro trabajo, lo que haremos será decir que dos palabras forman una pareja de palabras encadenadas si las dos últimas letras de una se corresponden con las dos primeras de otra en la lista. Ten en cuenta que una palabra puede formar pareja con varias palabras de la lista.

### Entrada

Por cada caso de prueba se recibirá un entero  $N$ , seguido de  $N$  palabras de longitud  $L$  que conforman la lista (siempre habrá un mínimo de 2 palabras en la lista).

La entrada debe ser leída de forma estándar.

### Salida

Para cada caso debemos imprimir el número de parejas de palabras encadenadas que hay en total en nuestra lista.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
3 hola lazo zocalo 6 cono laca laboratorio ola sala ionico 2 lana lazo	2 6 0

### Límites

- $2 \leq N \leq 100$
- $2 \leq L \leq 23$

# Problema B

## Tendencias en redes sociales

En las redes sociales, existen ahora ciertas figuras, llamados “Influencers”. Estas personas son atractivos para las marcas pues, como tienen un número siempre creciente de seguidores, pueden promocionar y por ende generarle muchas ganancias a dicha empresa.

Nuestra tarea de hoy es analizar un conjunto de personas y su evolución en Histogram, una red social que se basa en subir fotos de tus histogramas preferidos. Para cada persona, veremos su evolución a través de los últimos  $N$  días y si su evolución es siempre creciente, lo queremos contactar, de lo contrario, si pierde tan siquiera un seguidor, eligiémoslo no contactarlo. Puesto que no nos queremos arriesgar a dar un falso positivo.

### Entrada

La primera línea contiene un número  $T$  con los casos de prueba a evaluar.

Por cada caso de prueba recibirás dos enteros  $N$  y  $M$  denotando el número de días a evaluar y el número de personas a evaluar respectivamente, luego,  $2 * M$  líneas, primero con el nombre de la persona que estamos evaluando  $S$  (solo con mayúsculas y minúsculas del alfabeto inglés), luego,  $N$  números separados por un espacio denotando, desde hace  $N$  días hasta hoy, su evolución en número de seguidores en la red social.

La entrada debe ser leída de forma estándar.

### Salida

Para cada caso debes imprimir, por orden de aparición, los nombres de las personas a contactar separadas por un espacio, si no hace falta contactar a ninguna persona, imprime una línea vacía.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
3 2 2 Yoyo 100 101 VuelaPhone 100 100 2 2 Alberto 100 99 PalmerasMalas 2 2 3 4 Alberto 300 700 1000 Juan 300 100 250 PalmerasBuenas 1000 10000 100000 AlmacenesJacinto 100 100 100	Yoyo  Alberto PalmerasBuenas

## Límites

- $1 \leq T \leq 100$
- $2 \leq N \leq 100$
- $2 \leq M \leq 100$
- $1 \leq N_i \leq 1,000,000,000$
- $1 \leq S \leq 100$

# Problema C

## Próximo Año Bisiesto

A lo largo de los años diferentes investigaciones han llegado a la conclusión de que la tierra tarda 365,2524 días en dar un giro alrededor del sol, es por esto que se dice que cada 4 años febrero tiene 29 días en lugar de 28 y que cada 100 años esta regla se ignora (a menos que pasen 400 años en cuyo caso la regla se vuelve a aplicar), es decir, cada 4 años existe un año bisiesto a menos que el año sea divisible entre 100 y no entre 400. Para no entrar en polémica con calendarios anteriores, asumiremos que el año dado será mayor o igual a 1900.

Dado el año de hoy, ¿Puedes calcular el próximo año bisiesto?

### Entrada

La primera línea contiene un entero  $T$  denotando el número de casos de prueba. Cada caso de prueba contiene solo un número  $Y$  con el año actual.

La entrada debe ser leída de forma estándar.

### Salida

Para cada caso imprime un número con el siguiente año bisiesto siguiendo las reglas del calendario gregoriano.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
3	1904
1900	2012
2011	2020
2018	

### Límites

- $1 \leq T \leq 100$
- $1900 \leq Y \leq 10,000$

# Problema D

## Agencia de Espías

El gobierno tiene un programa especial de espionaje y quiere probarlo en una situación lo más cercana a la realidad posible.

La situación simulada consiste en tener un grupo rojo y un grupo azul de espías, encomendarle misiones a ambos colores y que se espíen entre ellos mismos. La idea central es que los rojos espíen a los azules y viceversa, mas sin embargo, si los rojos espían a los rojos ó los azules espían a los azules no tendría mucho sentido el programa.

Tu cometido es simple, dadas las misiones en forma de  $A$  a  $B$  siendo que  $A$  y  $B$  se espían mutuamente. ¿Puedes asignar a cada espía un color (azúl o rojo) de tal forma que nadie espíe a alguien del mismo color?

### Entrada

La primera línea de entrada contiene un número  $T$  con los casos de prueba.

Por cada línea tendremos dos números  $N$  y  $M$ , denotando el número  $N$  de espías y las  $M$  relaciones que existen entre ellos. Por cada una de las siguientes  $M$  líneas habrán dos enteros  $A$  y  $B$  denotando que dichos espías se espíarán mutuamente.

La entrada debe ser leída de forma estándar.

### Salida

Imprime SI si es posible asignar uno de los dos colores a cada espía de tal forma que no se espíen entre su mismo color ó NO en el caso contrario.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
3	SI
4 4	NO
1 2	SI
2 3	
3 4	
4 1	
4 6	
1 2	
2 3	
3 4	
4 1	
1 3	
2 4	
4 2	
1 2	
3 4	

### Límites

- $1 \leq T \leq 10$
- $1 \leq N \leq 100.000$
- $1 \leq M \leq \min(N*(N-1)/2, 100000)$

# Problema E

## Versiones de programas

En la informática se suelen diferenciar los distintos cambios y mejoras del software mediante números separados por, usualmente, un punto '.', cada uno de estos puntos suele denotar si la mejora afecta en menor o mayor manera la calidad del software en general.

Estamos interesados en, dada una lista desordenada de versiones de software, como sería su representación ordenada. Para comparar dos versiones  $X$  e  $Y$  separamos los números de la versión del programa y si el  $i$ -ésimo número de  $X$  es mayor al  $i$ -ésimo número de  $Y$  decimos que  $X$  es una versión más avanzada que  $Y$ , si en cambio es menor,  $Y$  sería mayor que  $X$ .

Por simplicidad, si  $X$  tiene menos "puntos" que  $Y$ , rellenar con ceros lo que reste (e.g. Si  $X = 1.0$  e  $Y = 1.0.1$ , se compararía con  $X = 1.0.0$ )

### Entrada

La primera línea contiene un entero  $T$  con los casos de prueba.

Por cada caso de prueba se leerá un número  $N$  y después, en  $N$  líneas, un string  $S$  con el nombre de las versiones del software disponible.

La entrada debe ser leída de forma estándar.

### Salida

Para cada caso imprime una lista ordenada separadas por un espacio de menor a mayor con las versiones de software del programa.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
2 3 1.0 2.0 1.5 4 1.0 1.1.1 1.1 53.0.12	1.0 1.5 2.0 1.0 1.1 1.1.1 53.0.12

### Límites

- $1 \leq N \leq 100$
- $1 \leq S \leq 1,000$