

# Problema A

## Contar Vocales

El titulo resume todo el problema, dado una línea de strings cuenta cuantas vocales ('a', 'e', 'i', 'o', 'u') contiene. Las mayúsculas también las debes contar.

### Input

La primera línea del INPUT comienza con un número M indicando el número de casos ( $1 \leq M \leq 100$ ), M líneas le siguen.

- o Cada línea tendrá una serie de strings (que contendrá caracteres (A-Z a-z 0-9) de longitud entre 1 y 100, inclusive

### Output

Deberá formatear su salida de la forma "Caso #N: R".

(Comillas para claridad) donde N es el número de caso y R es el total de vocales que se encuentran en la entrada.

<b>Input</b>  3 aeiou huahcoding owns gcjtc jk fml
<b>Output</b>  Caso #1: 5 Caso #2: 5 Caso #3: 0

# Problema B

## Moys Day

Dada una lista de eventos que **Moy** realizó en un día, determinar si es un día normal para Moy.

Un día normal para Moy es cuando existen al menos un "HON", un "POKER" o un "JEZZ" (comillas por claridad), si es un día normal para Moy retornar las veces que estos eventos aparecen en el día de Moy.

Si no es un día normal para Moy retornar "FML" (Sin las comillas).

### Input

La primera línea del INPUT comienza con un número M indicando el número de casos ( $1 \leq M \leq 50$ ), M líneas le siguen.

- Cada línea tendrá una lista de Strings los cuales representa los eventos que sucedieron en el día de Moy separados por un espacio en blanco:
  - o Cada String tendrá solo letras en mayúsculas 'A'-'Z'.
  - o Cada String tendrá entre 1 y 50 caracteres, inclusive.

### Output

Deberá formatear su salida de la forma "Caso #N: R".

(Comillas para claridad) donde N es el número de caso y R es el resultado explicado anteriormente.

Input	Output
4 HON POKER JEZZ HON POKER XTIT HONO JEZZ JEZZ JEZZ POKER POKKER GGE LA SAGELL EM	Caso #1: 3 Caso #2: 2 Caso #3: 4 Caso #4: FML

# Problema C

## Príncipe Consonante

Era la edad media del alfabeto, cuando las vocales eran perseguidas por que a pesar de haber pocas en el alfabeto eran muy usadas en las palabras, las consonantes estaban celosas de esto. Una vez que las consonantes veían a una o un grupo de ellas las rodeaban y las aniquilaban.

Tú eres el Príncipe consonante que llevará a cabo la masacre y tu tarea es eliminar las vocales si ellas se encuentran rodeadas (por ambos lados) por una o más consonantes en una palabra, es decir las vocales interiores.

Una vocal es interior si hay al menos una consonante a la izquierda y la derecha (no necesariamente juntas) en la misma palabra. Dado a que te dan el conjunto de palabras en forma normal tienes que retornar el conjunto de palabras después de la masacre.

Dios perdone tu código.

### Input

La primera línea del INPUT comienza con un número M indicando el número de casos ( $1 \leq M \leq 150$ ), M líneas le siguen.

- Cada línea tendrá una cadena representando un mensaje normal sin acortar que es un String que contiene desde 0 a 50 caracteres, inclusive, (A-Z, a-z) y espacios(" ").

### Output

Deberá formatear su salida de la forma "Caso #N: R".

(Comillas para claridad) donde N es el número de caso y R es el resultado.

Input	Output
4 No me mates El principe kuaek I like your style pee on me	Caso #1: No me mts Caso #2: El princpe kk Caso #3: I lke yr style Caso #4: pee on me

# Problema D

## HuaHContest Aforo

Un bonito día, al admin de HuaHCoding se le vino una pregunta mientras hacia hora... "¿cuantos participantes asistirán para el próximo concurso?", el mismo se contestó, "es una pregunta difícil, pues hay muchos que van tan solo si van sus amigos", por ejemplo el competidor **LSTR** participara si y tan sólo si el competidor **Ziklon** va. Ya que la amistad es mutua en los HuaHCoders, **Ziklon** solo irá si **LSTR** va. O por ejemplo hay 3 amigos de lima que cada uno participará si y tan solo si los otros 2 participan. Es decir existe una dependencia mutua.

El poderoso admin conoce a todos los potenciales participantes y tiene la lista de ellos y sus dependencias entre ellos. Ahora quiere hallar **K** (el numero de asistentes) de tal manera que todos los participantes asistan con sus amigos (mutualmente dependientes) el día del concurso.

Ud. recibirá una lista de dependencias **D** (delimitadas entre ellas por una coma) que describirá las dependencias entre los participantes, El elemento **i** de **D** es otra lista (delimitadas por un espacio) de los participantes de los cuales **i** depende.

Por ejemplo de la siguiente entrada:

```
{1 2,0,0,}
```

- El participante 0 necesita del 1 y 2 para ir.
- El participante 1 necesita del 0 para ir.
- El participante 2 necesita del 0 para ir.
- El participante 3 no necesita de nadie para ir.

El primer participante tiene índice 0, el segundo tiene índice 1 y así sucesivamente. Note el número de participantes es el número de comas + 1, por lo tanto el ultimo no tiene dependencias.

Tu tarea es responder una lista con todos los posibles valores positivos para **K** en orden ascendente.

### Input

La primera línea del INPUT comienza con un número M indicando el número de casos ( $1 \leq M \leq 65$ ), M líneas le siguen.



- Cada caso esta formado por una línea que representa la lista de dependencias **D** que contienen entre 1 y 30 (inclusive) elementos separadas por una coma, con el siguiente formato:
  - o {d0,d1,d2...}
  - Cada elemento **d** es una lista de índices separados por un espacio en blanco que contiene entre 0 y 50 caracteres.
  - Cada entero en **d** esta entre 0 y (el número de elementos - 1), inclusive, y no contiene ceros adelante (si no es el cero).
  - Cada elemento en **d** no contiene valores duplicados.
  - Si el elemento **i** de **D** contiene un valor **j**, entonces el elemento **j** de **D** contiene el valor **i**.
  - El Elemento **i** de **D** no contiene el valor de **i**;

## Output

Deberá formatear su salida de la forma "Caso #N: R".

(Comillas para claridad) donde N es el número de caso y R es la lista de todos los K posibles separados por un simple espacio.

### Input

```
4
{1 2,0,0,}
{1 2,0 2,0 1}
{,,}
{4 2,3,0 4,1,0 2,6,5}
```

### Output

```
Caso #1: 1 3 4
Caso #2: 3
Caso #3: 1 2 3 4
Caso #4: 2 3 4 5 7
```

Caso 2: Tenemos que tomar todos los participantes, no hay manera que valla uno solo o dos solos.

Caso 3: Podemos tomar cualquier número de esos participantes, ya que ellos no tienen ninguna dependencia