

Problema A

Justificar Texto

Tenemos una colección de Strings y queremos justificarlos todos a la derecha. Dado una colección de Strings tu tarea es retornar la misma colección pero justificados a la derecha en el mismo orden que ellos aparecen en la entrada. Deben agregar tantos espacios en blanco delante de cada String de tal manera que todos los Strings tengan la misma longitud que el String de mayor longitud.

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 cadena representando una colección de Strings los cuales sigue la siguiente configuración:
 - o {String1,String2,String3}
 - 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, Asegúrese de agregar las llaves que encierran a la colección.

Input

```
4
{BOB,TOMMY,JIM}
{JOHN,JAKE,ALAN,BLUE}
{INTERNATIONALIZATION}
{AL,BOB,JOHN,JIMMY,HAROLD,DOUGLAS,JONATHON}
```

Output

```
Caso #1: {   BOB,TOMMY,   JIM}
Caso #2: {JOHN,JAKE,ALAN,BLUE}
Caso #3: {INTERNATIONALIZATION}
Caso #4: {           AL,           BOB,           JOHN,           JIMMY,           HAROLD, DOUGLAS,JONATHON}
```

Problema B

Números Esperantos

Los números son más fáciles de escribir en Esperanto que en Español, Números del 1 al 10 están escritos como sigue: "unu", "du", "tri", "kvar", "kvin", "ses", "sep", "ok", "nau", "dek". Números del 11 al 19 son escritos: "dek unu", "dek du", ..., "dek nau" – un “dek” seguido por un simple espacio y el número para el ultimo dígito. Números del 20 al 29 están escritos: "dudek", "dudek unu", "dudek du", ..., "dudek nau". Similarmente, 30 es "tridek", ..., 90 es "naudek" – Solo juntamos los números de las decenas con “dek”. No hay excepciones como “doce” o “trece” en Español.

Dado un entero x entre 1 y 99, inclusive, retorna su escritura en Esperanto.

Input

La primera línea indica el número de casos a analizar. Cada caso está compuesto por una línea, conteniendo solo un entero X , representando el número del cual deberá ser resuelto ($1 \leq X \leq 99$).

Output

Para cada caso de prueba en el input, tu programa debe imprimir una línea simple, conteniendo “Caso #X: Y” donde X es el número del caso de prueba comenzando de 1, Y es el número escrito en Esperanto.

Input de ejemplo	Output del Input de ejemplo
4	Caso #1: unu
1	Caso #2: naudek
90	Caso #3: dek unu
11	Caso #4: kvindek sep
57	

Problema C

Mejorar Típeo

Channy quiere convertirse en un gran programador, así que él practica para mejorar sus habilidades de programación todo el tiempo. Él piensa que puede ahorrar mucho tiempo si aprende a tipear más rápido. Se sabe que la clave de la velocidad de tipeo es mantener el ritmo, lo que significa que se debe presionar cada tecla después de un intervalo de tiempo constante. Así que Channy creará un programa que para medir su progreso. El programa recibe una cadena y un arreglo de enteros, representando las teclas que él presionó y el tiempo cuando él las presiono en milisegundos respectivamente. El i -ésimo elemento del arreglo de enteros es el tiempo en el cual Channy presionó el i -ésimo carácter en la cadena. Ninguna tecla es presionada más de una vez. Y todos los tiempos son relativos al inicio de la sesión de práctica.

Tu tarea es calcular cuales teclas le toma a Channy mas tiempo que el tiempo promedio. Así él puede saber en cuales enfocarse más para practicar.

Para este problema tú puedes considerar el tiempo promedio como un entero (redondeado hacia abajo).

Input

La primera línea del INPUT comienza con un número M indicando el número de casos ($1 \leq M \leq 120$), $M \cdot 2$ líneas le siguen.

- Cada caso esta formado por 2 lineas:
 - o La primera línea del caso es la lista de caracteres que Channy presionó que contienen caracteres únicos todos en minúscula 'a'-'z' (entre 1 y 26 caracteres).
 - o La segunda línea es el arreglo de enteros (**tiempos**) indicando los tiempos en que Channy presiono cada tecla de la cadena correspondiente.
 - El tiempo que toma presionar el primer carácter es **tiempos[0]**, el tiempo que toma presionar el i -ésimo carácter es **tiempos[i]-tiempos[i-1]** para todo $i > 0$.
 - Cada elemento de **tiempos** esta entre 0 y 100000, inclusive.
 - Los elementos de **tiempos** están estrictamente en orden ascendente

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

```
3
dcab
250 300 400 800
keyboard
100 200 300 500 600 800 900 1200
rewq
500 1000 1500 4000
```

Output

```
Caso #1: db
Caso #2: bad
Caso #3: q
```

*Caso 1: los tiempos para d=250, c=50, a=100, b=400, el promedio es 200, entonces d y b son las letras que Channy necesita practicar más.

Problema D

SMS

Short message service (Servicio de Mensajería Corta), se ha convertido hoy en día en un método rápido para la comunicación. La mayoría de los proveedores de este servicio restringen el tamaño del mensaje, es por eso que se ha vuelto una necesidad acortar los mensajes. Los usuarios de teléfonos móviles han encontrado métodos para la compresión de sus mensajes de manera que el contenido de los mensajes se mantiene inalterado. Uno de ellos es llevar el mensaje original y eliminar todas las vocales interiores de cada palabra. Una vocal es interior si hay al menos una consonante a la izquierda y la derecha (no necesariamente pegadas) en la misma palabra. Dado a que te dan el mensaje en formato normal tienes que retornar el mensaje acortado.

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 Lets meet tomorrow Please come to my party I like your style pee on me	Caso #1: Lts mt tmrrw Caso #2: Plse cme to my prty Caso #3: I lke yr style Caso #4: pee on me

*Se están considerando los espacios en blanco.

Fibra Óptica

Un Nuevo tipo de fibra óptica inventado por el Optical Research Lab puede cambiar la manera en cual los datos es trasmitidos alrededor del mundo, llevando a los usuarios a otro nivel de internet. Esta fibra usa un nuevo material que puede incrementar la velocidad de la luz usando sus propiedades reflectivas, de esta manera nosotros podemos enviar un conjunto de rayos en el mismo cable usando diferentes "canales".

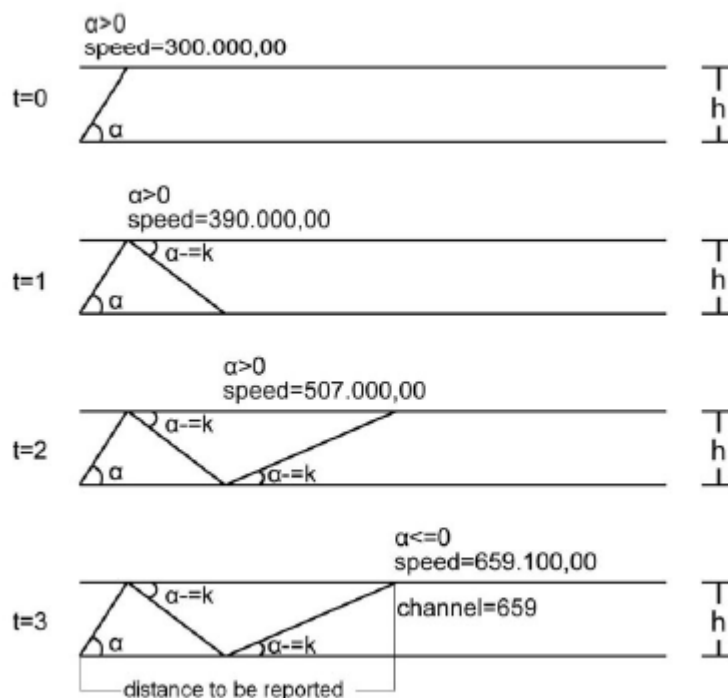
Funciona de esta manera: Un rayo generador envía rayos de luz desde la esquina inferior-izquierda del cable con un ángulo α , donde $0 < \alpha < 90$ (medido en grados, desde el fondo del cable). Con una velocidad inicial C .

Cada vez que un rayo rebota contra las paredes del cable, dos cosas ocurren:

- El ángulo α disminuye en k grados, $1 < k < 10$.
- La velocidad se incrementa p porcentaje.

Cuando $\alpha \leq 0$, el rayo se pone en una completa posición horizontal relativa al cable tomando uno de los 999 canales disponibles y continua viajando con la nueva velocidad alcanzada (velocidad máxima).

El canal en cual el rayo viajará por la fibra óptica corresponde a los primeros tres (3) dígitos de la velocidad máxima. La siguiente figura explica el proceso



Dado un conjunto de rayos con la velocidad inicial (C), El factor p y el correspondiente ángulo inicial. (α) y un factor k de cada rayo, tu debes buscar si el conjunto puede trabajar en el cable dado al mismo tiempo. Que significa que dos rayos no pueden quedar en el mismo canal cuando alcanzan la máxima velocidad.

Input

El input comienza con un entero correspondiente al numero M ($1 \leq M \leq 10$), de casos de prueba.

M casos de prueba siguen. Cada caso de prueba está conformado por un numero real h correspondiente al diámetro del cable en cms ($0.01 \leq h \leq 10$), seguido por un numero real representando la velocidad inicial de la luz (C – medido en km/sec, $10000 \leq C \leq 300000$) y un entero p indicando el porcentaje por el cual la velocidad se incrementa ($1 \leq p \leq 50$); finalmente, un entero N ($1 \leq N \leq 1000$) con el numero inicial de rayos, seguido por $2N$ números reales. Cada par de números se refiere al ángulo inicial del rayo α ($1 \leq \alpha \leq 90$) y el factor k ($1 \leq k \leq 10$).

Output

En caso el cable sea capaz de soportar el conjunto de ratos tu debes imprimir la distancia desde el cable inicial en cual cada rayo obtiene su velocidad máxima medido en cms. Y agregar la phrase “Cable and angles are perfect”. La precision debe se de dos (2) decimales. Si el cable no soporta el conjunto de rayos, la salida debe ser “Try different angles”.

Sample input	Output for the sample input
2	cm: 1044.47
8.80 300000 13	cm: 2867.51
5	cm: 3117.93
7.00 3.20	cm: 658.35
50.60 4.20	cm: 163.99
89.50 1.10	Cable and angles are perfect.
43.00 5.23	
34.76 9.9	Try different angles.
5.12 16483.59 30	
2	
71 6	
68 6	