

Problema A

Código: Prefijo

Un código prefijo es un conjunto de palabras en la que ninguna palabra es un prefijo de otra palabra en el conjunto. Una palabra V se dice que es un prefijo de una palabra W si W comienza con V .
*Se distinguen mayúsculas y minúsculas.

Una propiedad importante de los códigos de prefijo es que son únicamente descifrables. Los códigos prefijo son usados todos los días como por ejemplo los números de teléfono (cuando le llaman por error cuando marcan mal un número, este debido a que su número es un prefijo de otro número). En caso de que sea código prefijo retorne "Yes" como respuesta y en caso de que no sea retorne "No" acompañado del menor índice del prefijo encontrado, por ejemplo si el índice 2 es prefijo de otro posterior entonces la respuesta será: "No 2"

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 serie de cadenas de caracteres separadas por un espacio, la línea contiene de 1 hasta 50 cadenas inclusive, y cada cadena contiene desde 1 a 50 caracteres inclusive y solo contendrá letras (A-Z,a-z) y números (0-9).

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
5 trivial 10001 011 100 001 10 no nosy neighbors needed 1010 11 100 0 1011 No not	Caso #1: Yes Caso #2: No 2 Caso #3: No 0 Caso #4: Yes Caso #5: Yes

*caso 2: "100" (índice 2) y "10" (índice 4) son prefijos de "10001" y "10" es prefijo de "100", por lo tanto no es código prefijo además "100" es el prefijo con menor índice.

* En el caso 5 considere que se están diferenciando mayúsculas y minúsculas.

Problema B

Repeticiones

Dada una cadena, de entrada, con un máximo de 50 caracteres, encontrar la longitud de la subcadena más larga que aparece al menos dos veces (no se sobreponen) en la cadena. Si una subcadena no aparece dos veces, no se cuenta, devuelves 0.

Las cadenas son mayúsculas y minúsculas, y sólo letras mayúsculas y minúsculas se permiten en la cadena.

Por ejemplo, en la cadena "ABCDEXXXYYZZZABCDEZZZYYXX" la subcadena más larga que parece, al menos dos veces es "ABCDE". Estos dos subcadenas no se sobreponen por lo que la respuesta es 5.

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 de caracteres, la cadena contiene desde 1 a 50 caracteres inclusive y solo contendrá letras (A-Z,a-z).

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
5	
ABCDEXXXYYZZZABCDEZZZYYXX	Caso #1: 5
abcdabcdabcdabCD	Caso #2: 6
abcdefghijklmnopqrstuvwxyabcdefghijklmnopqrstuvwxy	Caso #3: 25
againANDagainANDagainANDagainANDagainANDagain	Caso #4: 21
abcdefghijklmnopqrstvwxyzABCDEFGHIJKLMNPOQRSTUVWXYZ	Caso #5: 0

* caso 2 "abcdab"+"cd"+"abcdab"+"CD" (se hace distinción de mayúsculas y minúsculas)

* caso 4 "againANDagainANDagain"+"AND"+"againANDagainANDagain"

Problema C

Estacionamiento

Te encuentras en una larga calle y necesitas estacionar tu auto, y para estacionarte tienes que cumplir los siguientes requerimientos:

1. No estar directamente en frente de un estacionamiento privado.
2. No estar directamente en frente de una parada de autobús.
3. No estar a 5 metros antes de una parada de autobús.
4. No estar a 10 metros antes de una parada de autobús.
5. No estar directamente en frente de una calle lateral.
6. No estar a 5 metros antes de una calle lateral.
7. No estar a 5 metros después de una calle lateral.

La calle se representa como una cadena de caracteres, donde cada carácter describe un tramo de la calle de 5 metros de longitud. Así el primer carácter describe los primeros 5 metros de la calle, el segundo carácter describe los próximos 5 metros y así sucesivamente. Para el estacionamiento privado se usara la letra "D", para la parada de autobús se usara la letra "B", para la calle lateral se usara la letra "S" y para los estacionamientos libres "-".

Una posición es directamente en "frente" si tiene el mismo índice, es "antes" si es un índice menos y es "después" si es un índice mas, por ejemplo te dan:

```

    " B - S "
Índices:  0 1 2

```

Para este caso la posición 0 está en frente de B y antes de -, la posición 1 está después de B, en frente de - y antes de S , y la posición 2 esta después de - y en frente de S.

Dada la cadena que representa la calle tú tendrás que devolver el número de estacionamientos libres en que puedes estacionarte.

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 de caracteres ,la cadena contiene desde 1 a 50 caracteres inclusive y solo contendrá las letras 'D', 'B', 'S' y '-'

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 ---B--S-D--S-- DDBDDDBDDDD --S--S--S--S-- SSD-B---BD-DDSB-----S-S-----S-B----BSB-S--B-S-D	Caso #1: 4 Caso #2: 0 Caso #3: 2 Caso #4: 14

Problema D

Palíndromas

Una palíndroma es una cadena de caracteres que se leen y escriben igual hacia adelante y hacia atrás. Queremos reordenar las letras de una cadena de caracteres que nos dan para convertirlo en una palíndroma.

Usted recibirá una cadena base de caracteres. Regresa el palíndromo que se puede hacer de dicha cadena base. Si se puede hacer más de un palíndroma, tienes que devolver el que es lexicográficamente antes (es decir, el que ocurra primero, en orden alfabético). Retorna "" (la cadena vacía) si no se puede hacer un palíndroma de la cadena base.

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 de caracteres ,la cadena contiene desde 1 a 50 caracteres inclusive y solo contendrá letras (A-Z).

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 AABB AAABB ABCD OTBOPTB	Caso #1: ABBA Caso #2: ABABA Caso #3: Caso #4: BOTPTOB

*Si se pueden hacer varios palíndromas se tiene que ordenar alfabéticamente.