

OCI PUC - 6

Agosto 2016

A. Reducción de números - NAME1

En un intento por reducir el crecimiento de la población, **Archer** ha sido consultado para elaborar un plan. **Archer** siendo tan inteligente como lo es, creó el siguiente plan:

Si N hijos, con nombres C_1, C_2, \dots, C_N nacen de padres con nombres **A** y **B**. Sea **C** la concatenación de todos los nombres de los hijos, es decir, $C = C_1 + C_2 + \dots + C_N$ (donde $+$ es el operador concatenación). Entonces **C** debe ser un substring de una de las permutaciones de **A + B**.

Tienes la tarea de verificar si los nombres que los padres quieren dar a sus hijos es permitido por el plan de **Archer** o no.

Input

La primera línea contiene un entero **T**, el número de casos de prueba. Siguen **T** casos de prueba. Cada prueba comienza (?) con dos strings **A** y **B** separados por un espacio, indicando los nombres de los padres. La siguiente línea contiene un único entero **N** indicando el número de hijos que **A** y **B** están planeando tener. Luego de esto hay **N** líneas, donde la **i**-ésima línea contiene **C_i**, el nombre propuesto para el **i**-ésimo hijo.

Output

Para cada caso de prueba una única línea que contiene "YES" si los nombres son permitidos por el plan de **Archer**, en caso contrario imprime "NO".

Restricciones

- $1 \leq T \leq 100$
- $1 \leq N \leq 1000$
- El largo de todos los strings, incluyendo **A**, **B** y todos los **C_i** estarán en el rango $[1, 40000]$, incluyendo ambos. Todos estos strings contendrán solo letras en inglés.

- Los largos combinados de todos los nombres de los hijos no excederán el largo combinado de los nombres de sus padres.

Ejemplo

Input:

```
3
tom marvoloriddle
2
lord
voldemort
cheap up
1
heapcup
bruce wayne
2
bat
man
```

Output:

```
YES
YES
NO
```

B. Lenguajes Olvidados - FRGTNLNG

Los lenguajes olvidados (también conocidos como lenguajes extintos) son lenguajes que ya no se usan. Tales lenguajes fueron, probablemente, muy usados antes y nadie podría haber imaginado que se extinguirían en algún punto. Desafortunadamente, eso es lo que les pasó. En el lado positivo, un lenguaje podrá estar muerto, pero algunas de sus palabras podrían continuar siendo usadas en otros lenguajes. Usando algo llamado *la internet*, has adquirido un diccionario de N palabras de un lenguaje olvidado. Mientras tanto, también sabes K frases usadas en lenguajes modernos. Para cada palabra del lenguaje olvidado, tu tarea es determinar si es que la palabra todavía está en uso en cualquiera de estas K frases modernas o no.

Input

La primera línea del input contiene un entero T que indica el número de casos de prueba. Le siguen la descripción de T casos:

La primera línea contiene dos enteros positivos N y K separados por un espacio.

La segunda línea contiene N strings que indican un diccionario del lenguaje olvidado.

Cada una de las siguientes K líneas comienza con un entero positivo L que indica el número de palabras en la frase correspondiente del lenguaje moderno. El entero es seguido por L strings (no necesariamente distintos) indicando la frase.

Output

Para cada caso de prueba, escribe una línea que contenga N tokens (separados por espacio): si la i -ésima palabra del diccionario existe en al menos una frase en los lenguajes modernos, entonces debes escribir **YES** como el i -ésimo token, de otra forma **NO**.

Restricciones

- $1 \leq T \leq 20$
- $1 \leq N \leq 100$
- $1 \leq K, L \leq 50$
- $1 \leq \text{largo de cualquier string en el input} \leq 5$

Ejemplo

Input :

2

3 2
piygu ezyfo rzotm
1 piygu
6 tefwz tefwz piygu ezyfo tefwz piygu
4 1
kssdy tjzhy ljzym kegqz
4 kegqz kegqz kegqz vxvyj

Output:
YES YES NO
NO NO NO YES

C. Número de factores - NUMFACT

Recientemente Alice ha aprendido a factorizar. Bob no cree que ella lo haya aprendido apropiadamente y entonces ha decidido hacerle una prueba. Bob le da a Alice un número muy grande y le pide encontrar el número de factores de este. Para hacerlo más fácil, él representa el número como el producto de N números. A Alice le asustan los números grandes y entonces te pide ayuda. Tu tarea es simple. Dados N números, debes decir el número de factores distintos del producto de estos N números.

Input

La primera línea del input contiene un único entero T , que representa los números de casos de prueba.

Cada prueba comienza con una línea que contiene un único entero N . La siguiente consiste en N enteros A_i separados por un espacio.

Output

Para cada caso de prueba, imprime en una línea separada el número total de factores del producto de los números dados.

Restricciones

- $1 \leq T \leq 100$
- $1 \leq N \leq 10$
- $2 \leq A_i \leq 1000000$

Ejemplo

Input :

```
3
3
3 5 7
3
2 4 6
2
5 5
```

Output :

```
8
10
3
```

Puntaje

- Obtendrás 40 puntos para resolver correctamente para $A_i \leq 100$.
- Obtendrás otros 30 puntos para resolver correctamente para $A_i \leq 10000$.
- Los 30 puntos restantes se obtienen al resolver correctamente para $A_i \leq 10000000$.

D. Akhil y String Aleatorio - ASTRING

Akhil se encuentra con un string **S** de largo **N**. Se comienza a preguntar sobre la subsecuencia lexicográficamente más pequeña de **S** de largo **K**.

Una subsecuencia de un string se forma al borrar algunos caracteres (es posible borrar 0) del string original.

Se dice que un string **A** es lexicográficamente menor que el string **B** del mismo largo si la primera posición donde **A** y **B** difieren, **A** contiene una letra que aparece antes en el diccionario que la letra correspondiente en **B**.

Input

La primera línea del input consiste en un entero **T** que indica el número de casos de prueba. Le sigue la descripción de **T** casos de prueba.

La primera línea de cada caso contiene un string **S**. La segunda línea de cada caso contiene un entero **K**.

Output

Para cada caso de prueba, imprime una única línea que contenga la subsecuencia de **S** más pequeña de largo **K**.

Restricciones

- $1 \leq T \leq 20$
- $1 \leq K \leq N$
- **S** consiste de caracteres en minúscula del alfabeto Inglés, es decir, de la 'a' a la 'z'.

Subtareas

- Subtarea #1 (10 puntos): $1 \leq |S| \leq 100$
- Subtarea #2 (20 puntos): $1 \leq |S| \leq 10^3$
- Subtarea #3 (70 puntos): $1 \leq |S| \leq 10^5$

Ejemplo

Input:

```
2
abdc
3
bacb
2
```

Output:

abc

ab

Explicación

Caso de ejemplo 1. `.abc.es` la subsecuencia lexicográficamente menor de entre `[.abd", "bdc", .abc", .adc"]`. Caso de ejemplo 2. `.ab.es` la subsecuencia lexicográficamente menor de largo 2.