

OCI PUC - 6

Agosto 2016

A. Limpiando - CLEANUP

Después de un largo y exitoso día de preparar comida para el banquete, es tiempo de limpiar. Hay una lista de n trabajos que hacer antes de que la cocina pueda cerrarse por la noche. Estos trabajos están indexados desde el 1 hasta n . La mayoría de los cocineros ya se fueron y solo el Chef y su asistente se quedaron a limpiar. Afortunadamente, alguno de los cocineros se preocupó de alguno de los trabajos antes de irse así que solo queda un subconjunto de los n trabajos. El Chef y su asistente se dividen los trabajos restantes de la siguiente manera: El Chef toma el trabajo sin terminar con el menor índice, el asistente toma el trabajo sin terminar con el segundo menor índice, el Chef toma el trabajo sin terminar con el tercer menor índice, etc. Los cocineros anotaron qué trabajos terminar antes de irse. Desafortunadamente, estos trabajos no fueron anotados en ningún orden en particular. Dada una lista no ordenada de trabajos sin terminar, debes determinar cuáles trabajos debe completar el Chef y cuáles su asistente antes de cerrar la cocina.

Input

La primera línea contiene un único entero $T \leq 50$ indicando el número de casos de prueba que siguen. Cada caso de prueba consiste en dos líneas. La primera línea contiene dos números n, m que satisfacen $0 \leq m \leq n \leq 1000$. Aquí, n es el total de trabajos que deben completarse antes de cerrar y m es el número de trabajos que ya se completaron. La segunda línea contiene una lista de m enteros distintos entre 1 y n . Esos son los índices de los trabajos que ya se completaron. Enteros consecutivos están separados por un espacio.

Output

El output de cada caso de prueba contiene dos líneas. La primera línea es una lista de los índices de los trabajos asignados al Chef. La segunda línea es una lista de los índices de los trabajos asignados a su asistente. Ambas listas deben aparecer en orden creciente y enteros consecutivos deben estar separados por un único espacio. Si el Chef o el asistente no tienen ningún trabajo asignado, entonces su línea correspondiente debe estar en blanco.

Ejemplo

Input :

3

6 3

2 4 1

3 2

3 2

8 2

3 8

Output :

3 6

5

1

1 4 6

2 5 7

B. Lapíndromos - LAPIN

Lapíndromo se define como un string tal que cuando se parte por la mitad, entrega dos mitades que tienen los mismos caracteres y la misma frecuencia para cada carácter. Si hay un número impar de caracteres entonces se ignora el carácter de al medio. Por ejemplo **gaga** es un lapíndromo, ya que las dos mitades **ga** y **ga** tienen los mismos caracteres con la misma frecuencia. También, **abccab**, **rotor** y **xyzxy** son unos pocos ejemplos de lapíndromos. Nota que **abbaab** NO es lapíndromo. Las 2 mitades contienen los mismos caracteres pero su frecuencia no calza. Tu tarea es simple. Dado un string, necesitas indicar si es un lapíndromo.

Input

La primera línea del input contiene un único entero **T**, el número de casos de prueba. Cada caso de prueba contiene un string **S** compuesto solo de letras del alfabeto inglés in minúscula.

Output

Para cada caso de prueba, escribe en una línea separada: "YES" si el string es lapíndromo y "NO" si no.

Restricciones

- $1 \leq T \leq 100$
- $2 \leq |S| \leq 1000$, donde $|S|$ denota el largo de **S**

Ejemplo

Input:

```
6
gaga
abcde
rotor
xyzxy
abbaab
ababc
```

Output:

```
YES
NO
YES
```

YES
NO
NO

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.