

C. Más Votado

Límite de tiempo: 1 segundos

Límite de memoria: 256 megabytes

Tras asegurar el equilibrio TCS en los textos, Paz dio el siguiente paso: construir un analizador de texto. Su nuevo programa debe estudiar la frecuencia de las palabras usadas por un grupo de personas.

Ella ha recibido muchos textos, por lo que hacer este analizador es importante. Sin embargo, no tiene tiempo porque está organizando la competencia *Codificadas*. Por eso te pide ayuda para hacer un programa que indique qué palabras dijo cada persona, determine cuáles fueron dichas por todas y muestre esas palabras ordenadas de la mayor a la menor frecuencia.

Entrada

La primera línea de la entrada consiste en un entero n ($1 \leq n \leq 100$), correspondiente a la cantidad de personas en el grupo a estudiar.

Por cada persona, la primera línea contendrá un entero p ($1 \leq p \leq 100$), correspondiente a la cantidad de palabras usadas por esa persona. Luego siguen p líneas, donde cada una contiene una palabra. Cada palabra está formada solo por letras minúsculas del alfabeto inglés, sin espacios, tildes ni acentos. Cada palabra puede tener un largo máximo de 100 caracteres.

Salida

La salida debe mostrar **las palabras que fueron usadas por todas las personas**. Escribe una palabra por línea.

El orden debe ser el siguiente:

1. **De mayor a menor frecuencia:** las palabras más usadas aparecen primero.
2. **En caso de empate de frecuencia,** se ordenan en **orden lexicográfico descendente**, es decir, la palabra lexicográficamente mayor aparece antes. Esto es parecido al orden de un diccionario, pero al revés: en lugar de que “abeja” aparezca antes que “zorro”, aquí “zorro” iría antes que “abeja”.

Ejemplos

Entrada 1	Salida 1
4 3 animal amigos casa 2 amigos casa 5 tcs codificadas amigos amigos casa 3 amigos amor casa	amigos casa

Entrada 2	Salida 2
1	tcs
5	come
tcs	codificadas
codificadas	
tcs	
come	
tcs	

Nota

En el primer ejemplo participan 4 personas:

- La primera usó 3 palabras: “animal”, “amigos” y “casa”.
- La segunda usó 2 palabras: “amigos” y “casa”.
- La tercera usó 5 palabras: “tcs”, “codificadas”, “amigos”, “amigos” y “casa”.
- La cuarta usó 3 palabras: “amigos”, “amor” y “casa”.

Las palabras usadas por todas las personas fueron “amigos” y “casa”. La palabra “amigos” apareció 5 veces y “casa” 4 veces. Por eso, “amigos” se muestra en la primera línea y “casa” en la segunda.

En el segundo ejemplo participa solo una persona, que utilizó 5 palabras. Como no hay más integrantes en el grupo, se considera que todas esas palabras fueron dichas por todas las personas. La palabra “tcs” apareció 3 veces, por lo que va en la primera línea. Las palabras “codificadas” y “come” aparecieron una vez cada una; como lexicográficamente “codificadas” es menor que “come”, “codificadas” va en la tercera línea y “come” en la segunda.

Definición de orden lexicográfico

El orden lexicográfico es el mismo orden en que los diccionarios organizan las palabras, y **la mayoría de los lenguajes de programación ya traen implementaciones listas para compararlas de esta forma.**

Dado un string a de longitud n y un string b de longitud m , se dice que a es lexicográficamente menor que b si se cumple alguna de las siguientes condiciones:

- **Primera diferencia:** existe una posición i en la que $a[i] \neq b[i]$, y el carácter en $a[i]$ es menor que el carácter en $b[i]$ según el alfabeto (o, en general, el valor de su código de carácter).

Ejemplo: “calabaza” es menor que “calazaza” porque coinciden en “cala”, pero en la posición 5 “b” es menor que “z”.

- **Prefijo:** Si todos los caracteres coinciden hasta que uno de los strings termina, entonces el string más corto es considerado lexicográficamente menor.

Ejemplo: “casa” es menor que “casamiento” porque “casa” aparece completo al inicio y tiene menos caracteres.