



EXAMEN FINAL

Programación Avanzada

Instrucciones para envío en Blackboard

Guarda tus archivos con el nombre según la siguiente regla:

- Nombre tus programas como <matrícula>pX.cpp, donde <matrícula>corresponde a los 6 dígitos de su matrícula UDEM y X al número de programa.
- Ejemplo: Si tu matrícula es 123456, el archivo para el problema 2, se deberá llamar M123456p2.cpp o .java
- Incluye el Código de Honor en cada programa.

*Resolver **INDIVIDUALMENTE** los siguientes problemas de programación. Evaluación acumulativa en 2 exámenes:*

Resolviendo 1: 60

Resolviendo 2: 90

Resolviendo 3: 110

Resolviendo 4: 125

Problema 1. *Videojuegos*

Standard input

Tiempo límite: 3 seg.

Pedro y Luis han decidido vender su enorme colección de videojuegos. Dado que no se quieren quedar sin sus preciados juegos, han decidido vender uno de cada uno de los títulos de los juegos que ambos poseen. ¿Puedes ayudar a Pedro y Luis a determinar cuántos juegos pueden vender?. Ni Pedro ni Luis poseen más de una copia de cada juego.

Input

La entrada consiste de una varios casos de prueba. La primer línea de cada caso contiene dos números enteros no negativos P y L , cada uno no mayor a 1 millón, que indica el número de videojuegos que poseen Pedro y Luis respectivamente. Después de esta línea le siguen P líneas que listan los números de catálogo de los juegos propiedad de Pedro en orden ascendente, y enseguida vendrán L líneas más que listan los números de catálogo de los juegos propiedad de Luis en ascendente. Cada número de catálogo es un número entero positivo no mayor de 1000 millones. La entrada termina con una línea con P y L igual a 0 (cero). Esta última línea no debe procesarse.

Output

Por cada caso de prueba, imprime una línea con un número entero que indica la cantidad de videojuegos que pueden vender Pedro y Luis.

Sample Input

```
1 2
7
7
9
3 3
1
2
3
1
2
4
0 0
```

Sample Output

```
1
2
```

Problem 2. *Contraseñas*

Standard Input

Time limit: 3 seg.

Algunos hackers tienen formas muy ingeniosas de crear sus contraseñas. Por ejemplo, Alejandro “el Hacker” Flores, oculta sus contraseñas de una forma ingeniosa. Alejandro elige una cadena S compuesta de letras minúsculas de longitud L . Luego, realiza corrimientos a la izquierda letra por letra de la cadena S y selecciona como contraseña la cadena que sea la primera lexicográficamente ordenadas de todas las cadenas obtenidas (incluyendo S).

Por ejemplo, si Alejandro utiliza la cadena “agenda”. Los cadenas generadas por los corrimientos letra por letra a la izquierda (incluida la cadena inicial) son:

```
agenda
gendaa
endaag
ndaage
daagen
aagend
```

En este caso, la contraseña elegida por Alejandro sería “aagend”. Observa que la primer letra de la contraseña elegida está en la posición 5 en la cadena inicial (las posiciones en la cadena se cuentan desde 0).

Tu trabajo consiste en escribir un programa que dada una cadena S encuentre la posición inicial de la primer letra de la contraseña elegida por Alejandro de una cadena S . Si la cadena elegida apareciera más de una vez, selecciona la posición inicial más pequeña.

Input

Tu programa debe leer varios casos de prueba. La primer línea contendrá un número C que indica la cantidad de casos de prueba. Cada una de las siguientes C líneas describirá un caso: primero la longitud L de la cadena ($5 \leq L \leq 100000$), seguido de un espacio simple, y seguido de la cadena S .

Output

Deberás imprimir exactamente C líneas, cada una con un solo número que indica la posición inicial de la primer letra de la contraseña elegida por Alejandro.

Sample Input

```
2
6 agenda
6 zaazaa
```

Sample Output

```
5
1
```