

A - N-sum

¡Ahora el ganso de uno está cocido y las zanahorias crudas están ahumadas!

El jefe de Per-Magnus irrumpió en su oficina y se quejó del programa adicional que había escrito para él anteriormente. ¡Solo puede sumar dos números, lo que por supuesto es completamente inutilizable! ¿Cómo se te ocurrió una idea tan tonta?

Arregle su programa rápidamente, para que en su lugar sume N enteros para un N dado.

Input

La primera línea de la entrada contiene un número entero N ($2 \leq N \leq 10$), el número de números enteros que debe agregar su programa.

La siguiente línea contiene los N enteros a sumar, cada uno entre 0 y 1000.

Output

Salida de un solo entero: la suma de los enteros de la entrada.

Ejemplo

Input

```
1 2
2 1 1
```

Output

```
1 2
```

Input

```
1 5
2 1 2 3 4 5
```

Output

```
1 15
```

B - Room Painting

Se está llevando a cabo una elección.

N personas votaron. La i -ésima persona ($1 \leq i \leq N$) emite un voto al candidato llamado S_i .

Encuentre el nombre del candidato que recibió más votos. La entrada proporcionada garantiza que hay un candidato único con la mayor cantidad de votos.

0.1. Restricciones

- $1 \leq N \leq 100$
- S_i es un string de largo entre 1 y 10 que consiste de solo letras minúsculas del abecedario.
- N es un entero
- Siempre existirá solo un candidato con más votos

Input

En la primera línea se encuentra un único entero N que consiste en la cantidad de votos emitidos. Luego, le siguen N líneas donde cada línea contiene el nombre de algún candidato.

Output

Muestre por pantalla el nombre del candidato que tuvo más votos.

Ejemplo

Input

```
1 5
2 snuke
3 snuke
4 takahashi
5 takahashi
6 takahashi
```

Output

```
1 takahashi
```

Input

```
1 5
2 takahashi
3 takahashi
4 aoki
5 takahashi
6 snuke
```

Output

```
1 takahashi
```

Input

```
1 1
```

```
2 a
```

Output

```
1 a
```

C - Shopping List

Kelly, siendo la gran programadora que es, con frecuencia piensa en "la próxima gran cosa" para las aplicaciones. Una mañana, Kelly pensó en la brillante idea de crear una red social para listas de compras. Kelly quiere hacer que su red social conecte a las personas que compran las mismas cosas en el supermercado, por lo que necesita su ayuda para crear un programa que averigüe cuándo los usuarios están comprando las mismas cosas en el supermercado para que su sitio pueda publicar anuncios relevantes. Kelly quiere comparar la última lista de compras de un usuario con todas sus listas de compras anteriores y averiguar qué artículos compran cada vez que van a la tienda.

Input

La primera línea de la entrada contendrá dos números enteros separados por espacios: n ($1 \leq n \leq 20$), el número de listas de compras, y m ($1 \leq m \leq 20$), el número de artículos en cada una de esas listas. Cada una de las siguientes n líneas es una de las listas de compras, cada una de las cuales contiene m artículos separados por espacios. Cada elemento tiene menos de 11 caracteres y está compuesto únicamente por caracteres ingleses en minúsculas.

Output

Muestra un número entero k , que indica el número de elementos que se encuentran en cada lista en la primera línea. En las siguientes líneas, muestre por pantalla estos elementos en orden alfabético, un elemento por línea.

Ejemplo

Input

```
1 4 5
2 bread carrots cheese eggs milk
3 milk carrots bread pasta yogurt
4 corn bread kale spinach carrots
5 milk bread cheese carrots yogurt
```

Output

```
1 2
2 bread
3 carrots
```

E - Batman v Superman

Batman preocupado por su inminente batalla con Superman intenta encontrar tantos fragmentos de kryptonita como sea posible. Recorrió el mundo y encontró S fragmentos de kryptonita.

En la Baticueva, solo hay una fila de compartimentos revestidos de plomo que la visión de rayos X de Superman no puede penetrar. Algunos de estos compartimentos ya están ocupados dejando solo N compartimentos libres.

Incluso en esta única fila, Batman decide mantener cada uno de los fragmentos lo más lejos posible entre sí. Tu tarea es ayudar a Batman a colocar los fragmentos de tal manera que la distancia más pequeña entre dos fragmentos sea la máxima posible. Encuentra y genera lo mismo.

Input

En la primera línea, se encuentra un entero N , número de compartimentos libres.

En la segunda línea, se encuentra un entero S , número de fragmentos.

En la tercera línea, se encuentra N enteros, las coordenadas de cada uno de estos compartimentos libres separados por un espacio.

Output

La distancia mínima máxima entre dos fragmentos.

Ejemplo

Input

```
1 5
2 3
3 1 4 2 9 8
```

Output

```
1 3
```

Explicación: puede mantener los fragmentos en 1, 4 y (8 o 9) dando la distancia mínima máxima a 3.