

Armando cajas de bombones*Contribución de Pablo Heiber***Descripción del problema**

Pedrito quiere regalar cajas de bombones a sus amigos y familiares, por lo que necesita preparar tantas cajas como sea posible.

Cada caja debe contener exactamente **K** bombones, **todos de tipo diferente**. Existen **N** tipos diferentes de bombones, y Pedrito dispone de x_i bombones de tipo **i** para cada $0 \leq i \leq N - 1$.

No es necesario que Pedrito utilice todos los bombones que tiene a su disposición, pero lógicamente un mismo bombón no puede utilizarse en más de una caja.

Dados **K** y las cantidades de bombones disponibles de cada uno de los **N** tipos existentes, debes escribir una función que calcule la máxima cantidad de cajas de bombones que Pedrito puede armar, cumpliendo con la restricción de no poner nunca dos bombones de un mismo tipo en la misma caja.

Descripción de la función

Debes implementar la función `bombones(K, x)`. Sus parámetros son:

- **K**: El entero **K** que indica la cantidad de tipos diferentes de bombones en cada caja.
- **x**: Un arreglo de **N** enteros, de modo tal que Pedrito tiene $x[i]$ bombones del **i**-ésimo tipo.

La función debe retornar un único entero: la máxima cantidad de cajas de bombones que Pedrito puede preparar.

Evaluador

El evaluador local lee de la entrada estándar con el siguiente formato:

- Enteros **N, K**
- Los **N** enteros x_i

El evaluador local escribe a la salida estándar la respuesta retornada por la función.

Restricciones

- $1 \leq K \leq N \leq 100.000$
- $1 \leq x_i \leq 10^9$

Ejemplo

Si se invoca al evaluador con:

```
5 3
1 2 10 3 4
```

La salida correcta es:

```
5
```

Si en cambio la entrada es:

```
3 1
10 15 20
```

La salida correcta es:

```
45
```

Subtareas

1. **N = 1** (4 puntos)
2. **K = 1** (10 puntos)
3. $x_i = 1$ para todo **i** (7 puntos)
4. **N ≤ 50, $x_i \leq 1000$** (20 puntos)
5. **N ≤ 10000, $x_i \leq 1000$** (30 puntos)
6. Sin más restricción (29 puntos)