

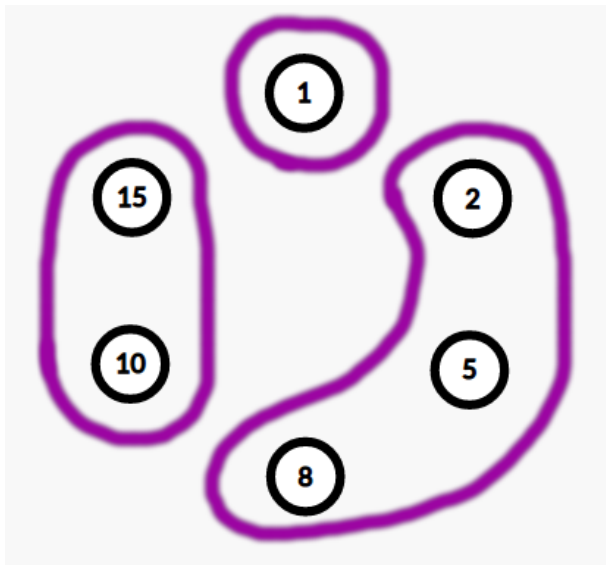
## Rompiendo la Ronda

Contribución de Agustín Santiago Gutiérrez

### Descripción del problema

Una ronda numérica es un círculo de números enteros positivos. Un *fragmento* de la ronda se puede obtener eligiendo uno o más números **consecutivos** de la ronda.

La siguiente figura muestra un ejemplo de ronda con  $N = 6$  números, dividida en  $K = 3$  fragmentos.



El *puntaje* de un fragmento se obtiene elevando al cuadrado el resto de dividir por 16 a la suma de los números del fragmento. Así, para los fragmentos del ejemplo, sus puntajes son:

- $(1 \bmod 16)^2 = 1$
- $((2 + 5 + 8) \bmod 16)^2 = 225$
- $((10 + 15) \bmod 16)^2 = 81$

El puntaje total se obtiene sumando los puntajes de todos los fragmentos, y en este ejemplo sería  $1 + 225 + 81 = 307$ .

Este no es el máximo puntaje posible con exactamente 3 fragmentos: por ejemplo, si se toman al **10** y al **15** como fragmentos de un único número, y el restante fragmento se forma con los números **1, 2, 5, 8**, se obtiene un puntaje total **325**. Si agregamos sin embargo la restricción de que el número **1** de este ejemplo debe ser utilizado en un fragmento de tamaño 1, entonces **307** es el mejor puntaje posible.

Debes escribir una función que dados los  $N$  números de la ronda y la cantidad  $K$  de fragmentos, calcule:

1. El máximo puntaje posible al separar la ronda en **exactamente**  $K$  fragmentos.
2. El máximo puntaje posible al separar la ronda en **exactamente**  $K$  fragmentos, pero con la restricción adicional que el primer número recibido debe ser utilizado para formar un fragmento que contiene únicamente ese primer número. Si ese mismo número apareciera muchas veces en la ronda, solamente esta primera aparición tiene esta restricción.

### Descripción de la función

Debes implementar la función `romperonda(ronda, k)`, que recibe:

- `ronda`: Un arreglo  $N$  enteros positivos. El primer elemento `ronda[0]` es el que está sujeto a la restricción especial para el cálculo del segundo puntaje.
- `k`: Entero que indica la cantidad de fragmentos en que debe dividirse la ronda.

La función debe retornar un arreglo con exactamente dos elementos: el máximo puntaje posible, y el máximo puntaje sujeto a la restricción adicional, en ese orden.

**Evaluador**

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

- En la primera línea, dos enteros **N** y **K**
- En la segunda línea, **N** enteros con los números del arreglo `ronda`.

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

**Cotas**

- $2 \leq K \leq N \leq 6.000$
- $K \leq 50$
- $1 \leq \text{ronda}[i] \leq 100$

**Ejemplo**

Si se invoca al evaluador con la siguiente entrada:

```
6 3
1 2 5 8 10 15
```

Para un programa correcto, la salida será:

```
325 307
```

En cambio para:

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

La salida será:

```
145 89
```

**Puntuación**

En cada caso de prueba, se obtiene el 50% del puntaje por calcular correctamente uno de los dos números pero no el otro.

**Subtareas**

1. **K = 2**: 8 puntos
2. **N ≤ 10**: 10 puntos
3. **N ≤ 100**: 22 puntos
4. **N ≤ 350**: 24 puntos
5. Sin más restricciones: 36 puntos