

Mordiendo porciones

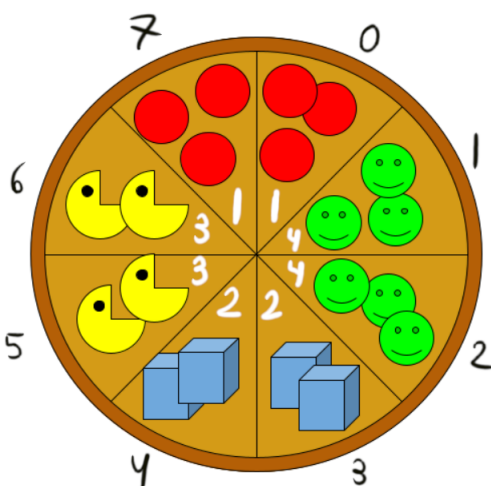
Contribución de Jeremías Figueiredo Paschmann

Descripción del problema

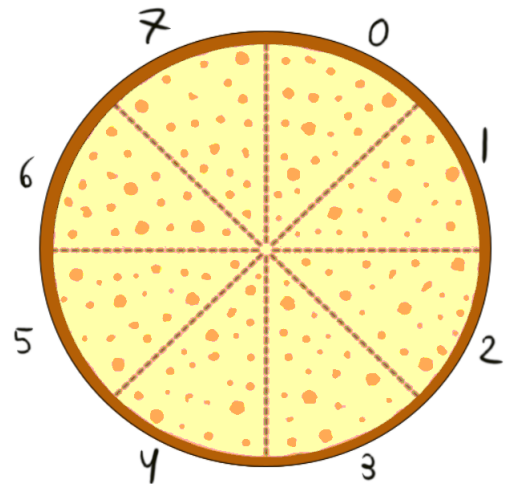
Las K personas que asisten al Pizza Camp Argentina 2025 llenaron un formulario antes de llegar con sus gustos favoritos de pizza. Con ello se construyó una enorme pizza de $N = K \cdot P$ porciones, de modo que a cada persona le toquen exactamente P porciones. A esta pizza de dimensiones récord se la conoce como *kpizza*.

Curiosamente, cada una de las K personas eligió un único gusto para todas sus porciones, y los K gustos son diferentes. Los gustos se numeran con enteros consecutivos desde 1 hasta K .

Los cocineros crearon la *kpizza* de forma tal que las P porciones de cada gusto están todas juntas, aunque los K gustos no están ordenados de ninguna manera particular en la *kpizza*. Las porciones de la *kpizza* están numeradas con enteros desde 0 hasta $N - 1$ inclusive, en sentido horario. La siguiente figura ilustra un ejemplo posible de *kpizza* con $N = 8$, $K = 4$ y $P = 2$. En el centro se indican en blanco los números correspondientes al gusto de cada porción, y en el exterior se indican en negro los identificadores de las porciones.



Los cocineros le pusieron demasiado queso a la *kpizza*, por lo cual al sacarla del horno estaba completamente cubierta y ahora es imposible distinguir a simple vista cuál es el gusto de cada porción.



Por la profunda cobertura de queso, la única forma de saber el gusto de cada porción es mordéndola. No se pueden realizar muchas mordidas porque las personas del Pizza Camp se van a indignar.

Debes implementar una estrategia para, sabiendo únicamente N , determinar la cantidad K de gustos distintos presentes en la *kpizza*. Tu puntuación será mayor cuantas menos mordidas se realicen (ver la sección de Puntuación).

Descripción de la función

Se debe implementar una función `kpizza(N)`. Sus parámetros son:

- N : Entero que indica la cantidad total de porciones de pizza.

La función debe retornar un único entero, que indique la cantidad de gustos distintos presentes en la *kpizza*.

En la implementación, se puede llamar a la función `morder(i)`, con $0 \leq i < N$ que indica el número de porción que se desea morder. La función `morder` retorna un único entero entre 1 y K , que indica el gusto de la porción i .

Notar que a diferencia del evaluador local, el comportamiento del evaluador en CMS es adaptativo: los gustos de las porciones pueden no estar prefijados al comenzar la ejecución de un cierto caso de prueba, sino que pueden depender de qué llamadas a `morder` se realicen en ese caso de prueba.

Evaluador local

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

- Un entero N
- N enteros que indican los sabores de las porciones (los que retornarán las llamadas `morder(i)`) para cada $0 \leq i < N$

Va escribiendo en la salida estándar las llamadas `morder(i)` que va realizando la función `kpizza`, y finalmente muestra el entero retornado por la función `kpizza`.

Restricciones

$$2 \leq N \leq 10^6$$

Ejemplos

Si la entrada es:

```
8
1 4 4 2 2 3 3 1
```

Ejemplos de interacciones correspondientes a una solución correcta podrían ser:

```
morder(0) = 1
morder(1) = 4
morder(2) = 4
morder(3) = 2
respuesta = 4
```

```
morder(1) = 4
morder(2) = 4
respuesta = 4
```

Puntuación

En cada caso de prueba, se obtiene un puntaje parcial p entre 0 y 1 de acuerdo a las siguientes fórmulas:

- $Q_{ex} = \max(0, Q - 5 - \lceil \lg N \rceil)$
- Si $0 \leq Q_{ex} < 30$: $p = \left(\frac{97}{100}\right)^{Q_{ex}}$
- Si $30 \leq Q_{ex} < 200$: $p = \frac{200 - Q_{ex}}{425}$
- Si $200 \leq Q_{ex}$: $p = 0$

donde Q es la cantidad de llamadas realizadas a la función `morder` y $\lg N$ es el logaritmo en base 2 de N . Por ejemplo, si en un caso de prueba $N = 10$ y se hacen $Q = 12$ mordidas, se tiene $Q_{ex} = \max(0, 12 - 5 - 4) = 3$ y el puntaje del caso es $p = \left(\frac{97}{100}\right)^3 = 0,912673$. En cambio con $Q = 47$ en ese mismo ejemplo se tendría $p = \frac{200 - 38}{425} = 0,381176$.

En una subtarea de valor T puntos, se obtiene un puntaje final $T \cdot p$ por esa subtarea, donde p es el **mínimo** puntaje obtenido en un caso de prueba de la subtarea. Si por ejemplo para una subtarea que vale $T = 5$ puntos el mínimo p entre sus casos de prueba es $p = 0,912673$, se obtienen 4,56 puntos por esa subtarea.

Subtareas

1. $N = 2$ (2 puntos)
2. $N = 6$ (3 puntos)
3. $N \leq 10$ (5 puntos)
4. $N \leq 3000$ (20 puntos)
5. N es potencia de 2 (16 puntos)
6. Sin más restricción (54 puntos)