

Breakdown Desastroso

Limite de tempo: 3s
Limite de memória: 256MB

Autor: Kattis - Disastrous Downtime

Você está investigando o que aconteceu quando um dos seus sistemas computacionais recentemente parou de funcionar. Até agora, concluiu que o sistema ficou sobrecarregado; aparentemente, ele não conseguiu lidar com a enxurrada de requisições recebidas. Desde o incidente, você teve amplas oportunidades de adicionar mais servidores ao seu sistema, o que o tornaria capaz de lidar com mais requisições simultâneas. No entanto, você simplesmente tem sido preguiçoso demais para fazer isso, até agora. Hora de por a mão na massa!

Para prever as futuras requisições ao seu sistema, você entrou em contato com os clientes do seu serviço, pedindo detalhes sobre como eles o utilizarão num futuro próximo. A resposta foi bastante impressionante; seus clientes enviaram uma lista com o instante exato de cada requisição que eles farão!

Você produziu uma lista de todas as requisições futuras, especificadas em milissegundos. Sempre que uma requisição chega, ela é imediatamente enviada a um dos seus servidores. Cada requisição leva exatamente 1000 milissegundos (1 segundo) para ser processada, e deve ser processada imediatamente.

Cada servidor pode trabalhar em, no máximo, k requisições simultaneamente. Dada essa limitação, calcule o número mínimo de servidores necessários para evitar outra falha no sistema.

Entrada

A primeira linha contém dois inteiros n e k , onde n ($1 \leq n \leq 100\,000$) representa o número de requisições futuras, e k ($1 \leq k \leq 100\,000$) representa o número máximo de requisições simultâneas que cada servidor pode processar.

Em seguida, seguem n linhas, cada uma contendo um inteiro t_i ($0 \leq t_i \leq 100\,000$), indicando que a i -ésima requisição ocorrerá t_i milissegundos após o momento exato em que você notificou seus clientes.

Os valores t_i estão em ordem cronológica, ou seja, $t_1 \leq t_2 \leq \dots \leq t_n$. É possível que várias requisições ocorram exatamente no mesmo instante.

Saída

A saída deve conter um único inteiro representando o número mínimo de servidores necessários para processar todas as requisições sem que o sistema fique sobrecarregado.

Exemplo

Entrada	Saída
2 1	1
0 1000	
3 2	2
1000 1010 1999	

Notas

No primeiro caso de teste, há duas requisições, e cada servidor pode processar apenas uma requisição por vez. As requisições ocorrem nos instantes 0 e 1000 milissegundos. Como cada requisição leva exatamente 1000 milissegundos para ser concluída, a segunda começa exatamente quando a primeira termina. Assim, não há sobreposição de processamento, sendo necessário apenas um servidor ativo para atender todas as requisições sem atraso.

No segundo caso de teste, há três requisições e cada servidor pode lidar com até duas requisições simultaneamente. As requisições ocorrem nos instantes 1000, 1010 e 1999 milissegundos. Dessa forma, é possível dividir o processamento entre dois servidores, sendo que um ficará responsável pelas requisições no instante 1000 e 1010 e o outro pela requisição no instante 1999.