

Problema A: O exército de clones de Palpatine (Divisão e Conquista)

O imperador Palpatine olhou para seu exército de clones e um belo problema surgiu em sua mente. Ele está te convidando a resolvê-lo.

Existem n clones no exército, dispostos em uma fila. Palpatine tem um telescópio, que ele usa para olhar os clones.

Inicialmente, Palpatine observa os clones no segmento $[1, n]$, e ele tem um valor da sorte de 0. Palpatine quer procurar o clone na posição do meio para cada segmento $[l, r]$ que ele observa. Então, o seguinte procedimento recursivo é usado:

- Primeiro, ele calculará $m = \lfloor (l + r) / 2 \rfloor$.
- Se o comprimento do segmento (ou seja, $r - l + 1$) for par, Palpatine o dividirá em dois segmentos de comprimento igual $[l, m]$ e $[m + 1, r]$ para observação adicional.
- Caso contrário, Palpatine apontará o telescópio para o clone m , e seu valor da sorte aumentará em m ; posteriormente, se $l \neq r$, Palpatine continuará a observar dois segmentos $[l, m - 1]$ e $[m + 1, r]$.

Palpatine é um pouco preguiçoso. Ele define sua preguiça por um número inteiro k : à medida que a observação avança, ele não continuará a observar nenhum segmento $[l, r]$ com um comprimento estritamente menor que k . Nesse caso, por favor, preveja seu valor da sorte final.

Entrada

Cada teste contém múltiplos casos de teste. A primeira linha da entrada contém um único inteiro t ($1 \leq t \leq 10^5$) — o número de casos de teste. A descrição dos casos de teste segue.

A única linha de cada caso de teste contém dois inteiros n e k ($1 \leq k \leq n \leq 2 \cdot 10^9$).

Saída

Para cada caso de teste, imprima um único inteiro — o valor da sorte final.

Exemplos

Input	Output
6	12
7 2	18
11 3	196
55 13	1975581
5801 6	958900
8919 64	38416403456028
8765432 1	

Nota

No primeiro caso de teste, no início, Palpatine observa $[1,7]$. Como $[1,7]$ tem um comprimento ímpar, ele aponta para o clone 4 e, portanto, aumenta seu valor da sorte em 4. Então, ele é dividido em 2 novos segmentos: $[1,3]$ e $[5,7]$. O segmento $[1,3]$ novamente tem um comprimento ímpar, então Palpatine aponta para o clone 2 e aumenta seu valor da sorte em 2. Em seguida, ele é dividido em 2 novos segmentos: $[1,1]$ e $[3,3]$, ambos com um comprimento menor que 2, então nenhuma observação adicional é realizada. Para o intervalo $[5,7]$, o progresso é semelhante e o valor da sorte eventualmente aumenta em 6. Portanto, o valor da sorte final é $4 + 2 + 6 = 12$.

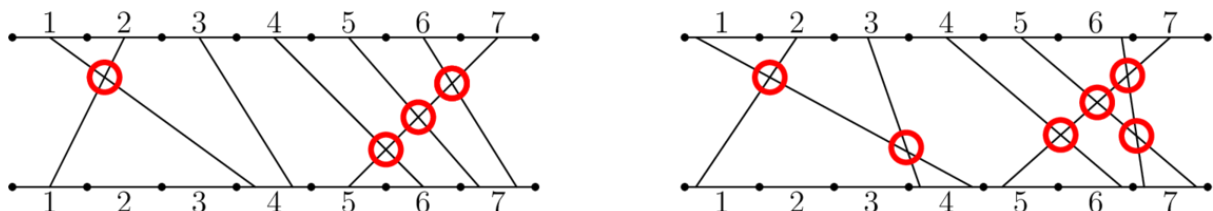
No último caso de teste, Palpatine finalmente observa todos os clones e o valor da sorte final é $1 + 2 + \dots + 8\,765\,432 = 38\,416\,403\,456\,028$.

Problema B: Conectando as placas

Um eletricista Francisco foi fazer um serviço na casa de um cliente, e parte do serviço era conectar duas placas eletrônicas através de fios. Ao realizar a conexão das placas, o eletricista percebeu que alguns dos fios se cruzaram, dessa forma ele se perguntou: "como eu posso conectar as duas placas de forma com que eu crie o máximo de cruzamentos de fios possível?"

Na imagem da esquerda, você pode ver a forma como o eletricista conectou os fios da primeira vez. Na imagem da direita, você pode ver como o eletricista conseguiu organizar os fios, de modo com que houvesse o máximo de cruzamentos de fios possíveis

:



Podemos perceber que todos os fios foram conectados nos mesmos locais nas duas imagens: os fios ligam do 1 ao 4, do 2 ao 1, do 3 ao 4, do 4 ao 6, do 5 ao 7, do 6 ao 7 e do 7 ao 5. Dessa forma, por mais que a organização dos fios mudaram, **eles continuam conectados no mesmo lugar**.

Portanto, ajude o eletricista a descobrir qual a quantidade máxima de cruzamento de fios possível quando for fazer a conexão de uma placa a outra.

Entrada

Podemos representar como as conexões devem ser feitas com um *array*: o índice i de cada posição do array indica qual porta será a origem do fio na placa eletrônica superior, e o valor a_i indica qual porta será o destino do fio na placa eletrônica inferior. Dessa forma, podemos representar o exemplo da imagem anterior em um *array* da seguinte forma:

[4, 1, 4, 6, 7, 7, 5]

Podemos ler então, por exemplo, o fio vai sair da conexão 1 da placa eletrônica superior para a conexão 4 da placa eletrônica inferior.

Assim, a primeira linha será um inteiro t que representa a quantidade de testes que serão executados. Esse número será maior ou igual a 1 e menor ou igual a 1000 ($1 \leq t \leq 1000$).

Ademais, a primeira linha de cada teste será um inteiro n que representará o tamanho do array. Esse inteiro poderá ser maior ou igual a 1 e menor ou igual a 100000 ($1 \leq n \leq 100000$).

Além disso, a próxima linha conterá n inteiros a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$). Estes são os elementos do array.

Saída

Para cada teste, a saída será um inteiro que representa a quantidade máxima de cruzamento de fios.

Exemplo

Entrada	Saída
4 7 4 1 4 6 7 7 5 2 2 1 1 1 3 2 2 2	6 1 0 3

Problema C: Problema dos waffles de McLovin (Programação Dinâmica)

Esta semana, McLovin decidiu fazer waffles e criar um problema relacionado a eles.

Ao preparar seu prato, decidiu preparar n waffles e, em seguida, colocá-los uns sobre os outros em uma pilha, adicionando um pouco de mel. McLovin começou com um prato vazio e realizou os seguintes passos n vezes:

- colocar um novo waffle no topo da pilha;

- após o i -ésima waffle colocado, despejar a_i colheres de mel no topo da pilha.

Quando x colheres de mel são despejadas no topo da pilha, os x waffles superiores são encharcados com o mel. Se houver menos de x waffles, todos os waffles são encharcados e o restante do mel é desperdiçado. Se $x=0$, nenhum waffle é encharcado.



A imagem representa o primeiro caso de teste do exemplo.

Ajude McLovin a determinar quais waffles na pilha acabam sendo encharcados quando o processo termina, e quais não são.

Entrada

Cada teste contém múltiplos casos de teste. A primeira linha contém o número de casos de teste t ($1 \leq t \leq 20\,000$). A descrição dos casos de teste segue.

A primeira linha de cada caso de teste contém um único inteiro n ($1 \leq n \leq 2 \cdot 10^5$) — o número de waffles.

A segunda linha de cada caso de teste contém n inteiros a_1, a_2, \dots, a_n ($0 \leq a_i \leq n$) — a quantidade de colheres de mel despejadas na pilha após adicionar cada waffle.

É garantido que a soma de n sobre todos os casos de teste não excede $2 \cdot 10^5$.

Saída

Para cada caso de teste, imprima uma única linha com n inteiros. O i -ésimo dos inteiros deve ser igual a 1 se o i -ésimo waffle de baixo para cima for encharcado, e 0 caso contrário.

Exemplos

Input	copy	Output	copy
<pre>3 6 0 3 0 0 1 3 10 0 0 0 1 0 5 0 0 0 2 3 0 0 0</pre>		<pre>1 1 0 1 1 1 0 1 1 1 1 1 0 0 1 1 0 0 0</pre>	

Problema D: Pedrinho e o campo minado

Rafael estava andando pelas florestas na Europa, quando de repente percebeu que ele estava em um campo minado antigo, da época das guerras que foram travadas no continente. Rafael então precisou chegar no final do campo minado sem ativar nenhuma das minas (ele conseguiu isso por conta que as minas estavam parcialmente visíveis). Dessa forma, ao chegar ao outro lado do campo sem pisar em nenhuma mina, Rafael decidiu deixar uma plaquinha mostrando os diversos caminhos que alguém poderia passar sem que essa pessoa pisasse em alguma mina.

Assim, Rafael desenhou um mapa de tamanho $n \times n$ onde o indivíduo só poderia se mover para a **direita** ou para **baixo**, e onde a pessoa entra pelo canto superior esquerdo e sai pelo canto inferior direito. Dessa forma, quantos caminhos Rafael conseguiu encontrar para ir do início do campo até ao fim dele sem pisar em nenhuma mina?

Entrada

A primeira linha da entrada é o tamanho n do campo minado: inteiro.

As próximas entradas são n linhas que contém . para indicar um caminho vazio, e * para indicar uma mina.

Restrições

A quantidade de caminhos possíveis x deve ser impressa usando módulo, para que o valor não dê overflow:

$x \bmod (10^9 + 7)$ - Resto da divisão de x por $(10^9 + 7)$.

n é um número entre 1 e 1000. Os dois extremos estão inclusos.

Saída

A saída é um número inteiro que indica a quantidade de caminhos em que uma pessoa poderia entrar no campo minado e sair sem ativar nenhuma mina.

Exemplo

Entrada	Saída
4*.. ...* *....	3

Nota

Ao guardar as informações na tabela de resultados pré-calculados, já o guardem usando $x \bmod (10^9 + 7)$.