# A. Математическая задача

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Ваш учитель по математике задал вам следующую задачку:

Есть $n$ отрезков на оси (прямой) $x$, $[l_1;r_1],[l_2;r_2],\ldots,[l_n;r_n]$. Отрезок $[l;r]$ включает свои границы, то есть это множество таких $x$, что $l \le x \le r$. Длина отрезка $[l;r]$ равна $r-l$.

Два отрезка $[a;b]$ и $[c;d]$ имеют общую точку (пересекаются), если найдется такое $x$, что $a \le x \le b$ и $c \le x \le d$. Например, отрезки $[2;5]$ и $[3;10]$ имеют общую точку, но отрезки $[5;6]$ и $[1;4]$ — не имеют.

Вам требуется добавить один отрезок, который имеет хотя бы одну общую точку с каждым данным отрезком, и имеет как можно меньшую длину. Искомый отрезок может вырождаться в точку (то есть быть отрезком длины ноль). Искомый отрезок может как быть, так и не быть среди заданных $n$ отрезков.

Другими словами, вам нужно найти такой отрезок $[a;b]$, что $[a;b]$ и $[l_i;r_i]$ имеют общую точку для всех $i$, и при этом значение $b-a$ минимально.

### Входные данные

В первой строке входных данных записано целое число $t$ ($1 \le t \le 100$) — количество наборов входных данных в тесте. Далее следуют описания $t$ наборов входных данных.

В первой строке набора входных данных содержится одно целое число $n$ ($1 \le n \le 10^5$) — количество отрезков. Следующие $n$ строк содержат описания отрезков — $i$-я из них содержит два целых числа $l_i, r_i$ ($1 \le l_i \le r_i \le 10^9$).

Сумма всех значений $n$ по всем наборам входных данных в тесте не превосходит $10^5$.

### Выходные данные

Для каждого из наборов входных данных выведите одно целое число — минимальную длину отрезка, который имеет хотя бы одну общую точку с каждым из данных отрезков.

**Пример**

входные данные

```
4
3
4 5
5 9
7 7
5
11 19
4 17
16 16
```

```
3 12
14 17
1
1 10
1
1 1
```

**выходные данные**

```
2
4
0
0
```

### Примечание

В первом примере, можно взять отрезок $[5;7]$ как ответ. Это самый короткий отрезок, который имеет хотя бы одну общую точку со всеми данными.

## B. Коробка

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

*Перестановка* $p$ — это последовательность целых чисел $p=[p_1,p_2,\ldots,p_n]$, которая состоит из $n$ различных положительных целых чисел от $1$ до $n$. Например, следующие последовательности являются перестановками — $[3,4,1,2]$, $[1]$, $[1,2]$. Следующие последовательности не являются перестановками — $[0]$, $[1,2,1]$, $[2,3]$, $[0,1,2]$.

Важный ключ спрятан в закрытой коробке, которую вам нужно открыть. Чтобы открыть коробку вам нужно ввести секретный ключ. Секретный ключ это перестановка $p$ длины $n$.

Эту перестановку вы не знаете, вы знаете только массив $q$ префиксных максимумов этой перестановки. Формально:

- $q_1=p_1$,
- $q_2=max(p_1,p_2)$,
- $q_3=max(p_1,p_2,p_3)$,
- ...
- $q_n=max(p_1,p_2,\ldots,p_n)$.

Вы хотите построить любую возможную исходную перестановку, которая согласуется с заданным массивом. Другими словами, найдите любую перестановку, что $q$ для этой перестановки равен данному массиву.

### Входные данные

В первой строке входных данных записано целое число $t$ ($1\leq t\leq 10^4$) — количество наборов входных данных в тесте. Далее следуют описания $t$ наборов входных данных.

В первой строке описания набора входных данных записано одно целое число $n$ $(1\leq n\leq 10^5)$ — количество элементов в перестановке-секретном коде $p$.

Во второй строке описания набора входных данных записаны $n$ целых чисел, $q_1$, $q_2,\ldots,q_n$ $(1\leq q_i\leq n)$: элементы массива $q$ для данной перестановки. Гарантируется, что $q_i\leq q_{i+1}$ для всех $i$ ($1\leq i< n$).

Сумма всех значений $n$ по всем наборам входных данных в тесте не превосходит $10^5$.

### Выходные данные

Для каждого набора входных данных выведите:

- Если невозможно найти подходящую перестановку $p$ выведите «-1» (без кавычек).
- Иначе, выведите $n$ различных целых чисел, $p_1, p_2, \ldots, p_n$ $(1 \leq p_i \leq n)$. Если для набора входных данных есть несколько возможных ответов, вы можете вывести любой.

### Пример

входные данные

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

выходные данные

```
1 3 4 5 2
-1
2 1
1
```

### Примечание

В первом наборе входных данных примера $[1,3,4,5,2]$ это единственный возможный ответ.

- $q_1=p_1=1$;
- $q_2=max(p_1,p_2)=3$;
- $q_3=max(p_1,p_2,p_3)=4$;
- $q_4=max(p_1,p_2,p_3,p_4)=5$;
- $q_5=max(p_1,p_2,p_3,p_4,p_5)=5$.

Можно доказать, что для второго набора входных данных примера не существует ответа.

## C. Грязно

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Вы устали от своей грязной комнаты, поэтому вы решили в ней прибраться.

Ваша комната это скобочная последовательность $s=s_1s_2\ldots s_n$ длины $n$. Каждый символ этой строки это либо открывающая круглая скобка '(', либо закрывающая круглая скобка ')'.

За одну операцию вы можете выбрать любую подстроку строки $s$ и перевернуть ее. Формально, за одну операцию вы можете выбрать подстроку $s[l\ldots r]=s_l,s_{l+1},\ldots,s_r$ и поменять порядок элементов в ней на $s_r,s_{r-1},\ldots,s_l$.

Например, если вы захотите перевернуть подстроку $s[2\ldots4]$ строки $s$=«(**(()**))», она станет равна $s$=«(**)(()**)».

**Правильной** скобочной последовательностью называется скобочная последовательность, которую можно преобразовать в корректное арифметическое выражение путем вставок между ее символами символов '1' и '+'. Например, скобочные последовательности «()()», «(())» — правильные (полученные выражения: «(1)+(1)», «((1+1)+1)»), а «)(» и «(» — нет.

Префиксом строки $s$ называется её подстрока, которая начинается с индекса $1$. Например, для строки $s$=«(())()» есть $6$ префиксов: «(», «((», «(()», «(())», «(())(» и «(())()».

По вашему мнению, красивая и чистая комната $s$ — это такая, что:

- вся строка $s$ целиком является *правильной скобочной*;
- **и** ровно $k$ ее префиксов (включая всю строку $s$) являются правильными скобочными.

Например, если $k=2$, то «(())()» это красивая и чистая комната.

Вы хотите использовать не более $n$ операций, чтобы сделать вашу комнату красивой и чистой. Операции применяются последовательно, одна за другой.

Гарантируется, что ответ существует. Обратите внимание, что вам **не нужно** минимизировать количество операций — найдите любой способ достичь требуемой конфигурации за $n$ или менее операций.

## Входные данные

В первой строке входных данных записано целое число $t$ ($1 \le t \le 100$) — количество наборов входных данных в тесте. Далее следуют описания $t$ наборов входных данных.

В первой строке записаны два целых числа $n$ и $k$ ($1 \le k \le \frac{n}{2}, 2 \le n \le 2000$, $n$ четно) — длина $s$ и необходимое количество правильных префиксов.

Во второй строке записана $s$ длины $n$ — данная скобочная последовательность. Все символы строки равны '(' или ')'.

Гарантируется, что в данной строке ровно $\frac{n}{2}$ символов '(' и ровно $\frac{n}{2}$ символов ')'.

Сумма всех значений $n$ по всем наборам входных данных в тесте не превосходит $2000$.

## Выходные данные

Для каждого из наборов входных данных выведите ответ.

В первой строке выведите число $m$ ($0 \le m \le n$) — количество операций. Вам **не требуется** минимизировать $m$, это число может быть любым.

В следующих $m$ строках выведите описания операций, каждая строка должна содержать два целых числа $l, r$ ($1 \le l \le r \le n$), описывающих операцию переворота подстроки $s[l \dots r]=s_l s_{l+1} \dots s_r$. Операции выполняются последовательно одна за другой.

Итоговая строка $s$ после применения всех операций должна быть правильной, ровно $k$ её префиксов (включая $s$) должны быть правильными.

Гарантируется, что ответ существует. Если есть несколько возможных ответов, вы можете вывести любой.

**Пример**

входные данные

```
4
8 2
()(())()
10 3
))()()()((
2 1
()
2 1
)(
```

выходные данные

```
4
3 4
1 1
5 8
2 2
```

```
3
4 10
1 4
6 7
0
1
1 2
```

**Примечание**

В первом примере итоговая последовательность — это «`()(()())`», в которой два префикса корректные, «`()`» и «`()(()())`». Обратите внимание, что все операции кроме «`5 8`» в примере вывода не меняют текущую строку $s$.

# D1. Оптимальные подпоследовательности (упрощённая версия)

ограничение по времени на тест: 3 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

*Это упрощенная версия задачи, в этой версии $1 \le n, m \le 100$. Вы можете взламывать эту задачу, только если вы заблокировали обе версии.*

Пусть задана последовательность целых чисел $a=[a_1, a_2, \dots, a_n]$ длины $n$. Её *подпоследовательностью* называется любая такая, которая получена удалением нуля или более элементов из последовательности $a$ (они не обязательно идут подряд). Например, для последовательности $a=[11,20,11,33,11,20,11]$:

- $[11,20,11,33,11,20,11]$, $[11,20,11,33,11,20]$, $[11,11,11,11]$, $[20]$, $[33,20]$ — подпоследовательности (некоторые из большого списка);
- $[40]$, $[33,33]$, $[33,20,20]$, $[20,20,11,11]$ — не являются подпоследовательностями.

Пусть дополнительно задано целое неотрицательное число $k$ ($1 \le k \le n$), тогда подпоследовательность называется *оптимальной*, если:

- она имеет длину $k$ и сумма её элементов максимальная возможная среди всех подпоследовательностей длины $k$;
- и среди всех подпоследовательностей длины $k$, которые удовлетворяют предыдущему пункту она *лексикографически* минимальна.

Напомним, что последовательность $b=[b_1, b_2, \dots, b_k]$ лексикографически меньше последовательности $c=[c_1, c_2, \dots, c_k]$, если первый слева элемент в котором они различаются меньше в последовательности $b$ чем в $c$. Формально: существует такое $t$ ($1 \le t \le k$), что $b_1=c_1$, $b_2=c_2$, ..., $b_{t-1}=c_{t-1}$ и одновременно $b_t<c_t$. Например:

- $[10,20,20]$ лексикографически меньше чем $[10,21,1]$,
- $[7,99,99]$ лексикографически меньше чем $[10,21,1]$,
- $[10,21,0]$ лексикографически меньше чем $[10,21,1]$.

Вам задана последовательность $a=[a_1, a_2, \dots, a_n]$ и $m$ запросов, каждый состоит из двух чисел $k_j$ и $pos_j$ ($1 \le k \le n$, $1 \le pos_j \le k_j$). Для каждого запроса выведите значение, которое стоит в индексе $pos_j$ оптимальной подпоследовательности заданной последовательности $a$ для $k=k_j$.

Например, если $n=4$, $a=[10,20,30,20]$, $k_j=2$, тогда оптимальная подпоследовательность равна $[20,30]$ — она минимальная лексикографически среди всех подпоследовательностей длины $2$ с максимальной суммой элементов. Таким образом, ответом на запрос вида $k_j=2$, $pos_j=1$ будет число $20$, а ответом на запрос вида $k_j=2$, $pos_j=2$ будет число $30$.

### Входные данные

В первой строке записано целое число $n$ ($1 \le n \le 100$) — длина последовательности $a$.

Вторая строка содержит элементы последовательности $a$ — целые числа $a_1, a_2, \ldots, a_n$ ($1 \le ai \le 10^9$).

Третья строка содержит целое число $m$ ($1 \le m \le 100$) — количество запросов.

Следующие $m$ строк содержат пары целых чисел $k_j$ и $pos_j$ ($1 \le k \le n$, $1 \le pos_j \le k_j$) — параметры запросов.

### Выходные данные

Выведите $m$ целых чисел $r_1, r_2, \ldots, r_m$ ($1 \le r_j \le 10^9$) по одному в строке — ответы на запросы в порядке их следования во входных данных.

Значение $r_j$ должно быть равно значению, которое содержится в позиции $pos_j$ оптимальной подпоследовательности для $k=k_j$.

**Пример**

входные данные

```
3
10 20 10
6
1 1
2 1
2 2
3 1
3 2
3 3
```

выходные данные

```
20
10
20
10
20
10
```

входные данные

```
7
1 2 1 3 1 2 1
9
2 1
2 2
3 1
3 2
3 3
1 1
7 1
7 7
7 4
```

выходные данные

```
2
3
2
3
2
3
1
1
3
```

**Примечание**

В первом примере, для $a = [10,20,10]$ оптимальные подпоследовательности:

- для $k=1$ — это $[20]$,
- для $k=2$ — это $[10,20]$,
- для $k=3$ — это $[10,20,10]$.

# D2. Оптимальные подпоследовательности (усложненная версия)

ограничение по времени на тест: 3 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

*Это усложненная версия задачи, в этой версии $1 \le n,m \le 2 \cdot 10^5$. Вы можете взламывать эту задачу, если вы ее заблокировали. Но вы можете взламывать предыдущую версию задачи, только если вы заблокировали обе версии.*

Пусть задана последовательность целых чисел $a=[a_1,a_2,\ldots,a_n]$ длины $n$. Её *подпоследовательностью* называется любая такая, которая получена удалением нуля или более элементов из последовательности $a$ (они не обязательно идут подряд). Например, для последовательности $a=[11,20,11,33,11,20,11]$:

- $[11,20,11,33,11,20,11]$, $[11,20,11,33,11,20]$, $[11,11,11,11]$, $[20]$, $[33,20]$ — подпоследовательности (некоторые из большого списка);
- $[40]$, $[33,33]$, $[33,20,20]$, $[20,20,11,11]$ — не являются подпоследовательностями.

Пусть дополнительно задано целое неотрицательное число $k$ ($1 \le k \le n$), тогда подпоследовательность называется *оптимальной*, если:

- она имеет длину $k$ и сумма её элементов максимальная возможная среди всех подпоследовательностей длины $k$;
- и среди всех подпоследовательностей длины $k$, которые удовлетворяют предыдущему пункту она *лексикографически* минимальна.

Напомним, что последовательность $b=[b_1,b_2,\ldots,b_k]$ лексикографически меньше последовательности $c=[c_1,c_2,\ldots,c_k]$, если первый слева элемент в котором они различаются меньше в последовательности $b$ чем в $c$. Формально: существует такое $t$ ($1 \le t \le k$), что $b_1=c_1$, $b_2=c_2$, ..., $b_{t-1}=c_{t-1}$ и одновременно $b_t<c_t$. Например:

- $[10,20,20]$ лексикографически меньше чем $[10,21,1]$,
- $[7,99,99]$ лексикографически меньше чем $[10,21,1]$,
- $[10,21,0]$ лексикографически меньше чем $[10,21,1]$.

Вам задана последовательность $a=[a_1,a_2,\ldots,a_n]$ и $m$ запросов, каждый состоит из двух чисел $k_j$ и $pos_j$ ($1 \le k \le n$, $1 \le pos_j \le k_j$). Для каждого запроса выведите значение, которое стоит в индексе $pos_j$ оптимальной подпоследовательности заданной последовательности $a$ для $k=k_j$.

Например, если $n=4$, $a=[10,20,30,20]$, $k_j=2$, тогда оптимальная подпоследовательность равна $[20,30]$ — она минимальная лексикографически среди всех подпоследовательностей длины $2$ с максимальной суммой элементов. Таким образом, ответом на запрос вида $k_j=2$, $pos_j=1$ будет число $20$, а ответом на запрос вида $k_j=2$, $pos_j=2$ будет число $30$.

### Входные данные

В первой строке записано целое число $n$ ($1 \le n \le 2 \cdot 10^5$) — длина последовательности $a$.

Вторая строка содержит элементы последовательности $a$ — целые числа $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Третья строка содержит целое число $m$ ($1 \le m \le 2 \cdot 10^5$) — количество запросов.

Следующие $m$ строк содержат пары целых чисел $k_j$ и $pos_j$ ($1 \le k \le n$, $1 \le pos_j \le k_j$) — параметры запросов.

### Выходные данные

Выведите $m$ целых чисел $r_1, r_2, \ldots, r_m$ ($1 \le r_j \le 10^9$) по одному в строке — ответы на запросы в порядке их следования во входных данных.

Значение $r_j$ должно быть равно значению, которое содержится в позиции $pos_j$ оптимальной подпоследовательности для $k=k_j$.

**Пример**

входные данные

```
3
10 20 10
6
1 1
2 1
2 2
3 1
3 2
3 3
```

выходные данные

```
20
10
20
10
20
10
```

входные данные

```
7
1 2 1 3 1 2 1
9
2 1
2 2
3 1
3 2
3 3
1 1
7 1
7 7
7 4
```

выходные данные

```
2
3
2
3
2
3
1
1
3
```

### Примечание

В первом примере, для $a=[10,20,10]$ оптимальные подпоследовательности:

- для $k=1$ — это $[20]$,
- для $k=2$ — это $[10,20]$,
- для $k=3$ — это $[10,20,10]$.

## E. Поджог в лесу Берляндии

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Представим лес Берляндии как бесконечное клеточное поле. В каждой клетке растет дерево. Или росло, до недавних событий.

Разрушительный огонь бушевал в лесу и уничтожил несколько деревьев. У вас есть прямоугольная карта $n \times m$, которая описывает поврежденную часть леса. Сожженные деревья отмечены на карте как «X», в то время как остальные отмечены как «.». **Вы можете быть уверены, что все поврежденные деревья отмечены на карте. Все деревья за пределами карты — не повреждены.**

Пожарные быстро потушили огонь, и теперь расследуют данный инцидент. Основная версия — это поджог: в какой-то момент времени (примем этот момент за $0$) несколько деревьев были подожжены. В начале минуты $0$ горели только подожженные деревья. В конце каждой минуты огонь распространяется от горящего дерева к $8$ соседним к нему деревьям. В начале минуты $T$ огонь был потушен.

Пожарные хотят найти поджигателей так быстро как это возможно. Но проблема в том, что они не знают ни значение $T$ (как долго бушевал пожар), ни местоположение деревьев, которые были подожжены. Они хотят, чтобы вы определили максимальное значение $T$ (чтобы узнать, сколько времени в запасе у поджигателей) и возможное множество деревьев, которые были подожжены.

Заметим, что вы хотите максимизировать значение $T$, но множество деревьев может быть произвольным.

### Входные данные

В первой строке заданы два целых числа $n$ и $m$ ($1 \leq n, m \leq 10^6$, $1 \leq n \cdot m \leq 10^6$) — размеры карты.

В следующих $n$ строках задана карта. Строка $i$ соответствует $i$-й строке карты и содержит строку из $m$ символов. $j$-й символ $i$-й строки либо «X», если соответствующее дерево сгорело, либо «.» — если осталось в целости.

Гарантируется, что карта содержит хотя бы один «X».

### Выходные данные

В первой строке выведите единственное число $T$ — максимальное время, которое мог гореть лес. В следующих $n$ строках выведите сертификат: карту (прямоугольник $n \times m$), на которой подожженные деревья отмечены как «X», а оставшиеся как «.».

### Пример

входные данные

```
3 6
XXXXXX
XXXXXX
XXXXXX
```

выходные данные

```
1
......
.X.XX.
......
```

**входные данные**

```
10 10
.XXXXXX...
.XXXXXX...
.XXXXXX...
.XXXXXX...
.XXXXXXXX.
...XXXXXX.
...XXXXXX.
...XXXXXX.
...XXXXXX.
..........
```

**выходные данные**

```
2
..........
..........
...XX.....
..........
..........
..........
.....XX...
..........
..........
..........
```

**входные данные**

```
4 5
X....
..XXX
..XXX
..XXX
```

**выходные данные**

```
0
X....
..XXX
..XXX
..XXX
```

## F1. Неправильный ответ на тесте 233 (упрощенная версия)

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Ваша программа снова не работает, на этот раз она получает вердикт «`Неправильный ответ на тесте 233`»

*Это упрощенная версия задачи, в этой версии $1 \le n \le 2000$. Вы можете взламывать эту задачу, только если вы заблокировали обе версии.*

В задаче идёт речь о тесте из $n$ вопросов. Каждый вопрос содержит $k$ вариантов ответа, и только один из них правильный. Ответ на $i$-й вопрос это $h_i$. Если ваш ответ на вопрос $i$ равен $h_i$, вы получите $1$ балл, в противном случае вы получите $0$ баллов за этот вопрос. В этой задаче значения $h_1, h_2, \dots, h_n$ вам известны (заданы).

Однако, вы допустили ошибку в вашей программе! Расположим все $n$ ответов на окружности по часовой стрелке. Из-за ошибки в вашей программе, они сдвигаются на один по циклу в направлении часовой стрелки.

Формально, ошибка двигает ответ на вопрос $i$ к вопросу $i \bmod n+1$. Так она двигает ответ на вопрос $1$ к вопросу $2$, ответ на вопрос $2$ к вопросу $3$, ..., ответ на вопрос $n$ к вопросу $1$.

Назовем все $n$ ответов вместе *набором ответов*. Всего есть $k_n$ возможных наборов ответов.

Вас интересует количество наборов ответов удовлетворяющих следующему условию: *после сдвига по часовой стрелки на 1, итоговое количество баллов нового набора ответов строго больше чем старого*. Вам необходимо найти это количество по модулю $998\,244\,353$.

Например, если $n=5$ и ваш набор ответов $a=[1,2,3,4,5]$, он будет изменен вашей программой на $a'=[5,1,2,3,4]$ из-за ошибки. Если набор правильных ответов равен $h=[5,2,2,3,4]$, тогда набор ответов $a$ получит $1$ балл, а набор ответов $a'$ получит $4$ балла. Так как $4>1$, набор ответов $a=[1,2,3,4,5]$ удовлетворяет условию и должен быть посчитан.

### Входные данные

В первой строке записаны два целых числа $n$, $k$ ($1 \le n \le 2000$, $1 \le k \le 10^9$) — количество вопросов и количество вариантов ответа в каждом вопросе.

Во второй строке записаны $n$ целых чисел $h_1, h_2, \ldots, h_n$, ($1 \le h_i \le k)$ — правильные ответы на вопросы.

### Выходные данные

Выведите одно целое число — количество наборов ответов, удовлетворяющих данному ограничению, по модулю $998244353$.

### Примеры

**входные данные**

```
3 3
1 3 1
```

**выходные данные**

```
9
```

**входные данные**

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

**выходные данные**

```
1000
```

### Примечание

На первый пример, корректные наборы ответов — это $[2,1,1],[2,1,2],[2,1,3],[3,1,1],[3,1,2],[3,1,3],[3,2,1],[3,2,2],[3,2,3]$.

## F2. Неправильный ответ на тесте 233 (усложненная версия)

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Ваша программа снова не работает, на этот раз она получает вердикт «`Неправильный ответ на тесте 233`»

*Это усложненная версия задачи, в этой версии* $1 \le n \le 2\cdot 10^5$. *Вы можете взламывать эту задачу, если вы ее заблокировали. Но вы можете взламывать предыдущую версию задачи, только если вы заблокировали обе версии.*

В задаче идёт речь о тесте из $n$ вопросов. Каждый вопрос содержит $k$ вариантов ответа, и только один из них правильный. Ответ на $i$-й вопрос это $h_i$. Если ваш ответ на вопрос $i$ равен $h_i$, вы получите $1$ балл, в противном случае вы получите $0$ баллов за этот вопрос. В этой задаче значения $h_1$, $h_2, \ldots$, $h_n$ вам известны (заданы).

Однако, вы допустили ошибку в вашей программе! Расположим все $n$ ответов на окружности по часовой стрелке. Из-за ошибки в вашей программе, они сдвигаются на один по циклу в направлении часовой стрелки.

Формально, ошибка двигает ответ на вопрос $i$ к вопросу $I \bmod n+1$. Так она двигает ответ на вопрос $1$ к вопросу $2$, ответ на вопрос $2$ к вопросу $3$, ..., ответ на вопрос $n$ к вопросу $1$.

Назовем все $n$ ответов вместе *набором ответов*. Всего есть $k_n$ возможных наборов ответов.

Вас интересует количество наборов ответов удовлетворяющих следующему условию: *после сдвига по часовой стрелки на $1$, итоговое количество баллов нового набора ответов строго больше чем старого*. Вам необходимо найти это количество по модулю $998\ 244\ 353$.

Например, если $n=5$ и ваш набор ответов $a=[1,2,3,4,5]$, он будет изменен вашей программой на $a'=[5,1,2,3,4]$ из-за ошибки. Если набор правильных ответов равен $h=[5,2,2,3,4]$, тогда набор ответов $a$ получит $1$ балл, а набор ответов $a'$ получит $4$ балла. Так как $4>1$, набор ответов $a=[1,2,3,4,5]$ удовлетворяет условию и должен быть посчитан.

## Входные данные

В первой строке записаны два целых числа $n$, $k$ ($1 \le n \le 2 \cdot 10^5$, $1 \le k \le 10^9$) — количество вопросов и количество вариантов ответа в каждом вопросе.

Во второй строке записаны $n$ целых чисел $h_1, h_2, \ldots, h_n$, ($1 \le h_i \le k$) — правильные ответы на вопросы.

## Выходные данные

Выведите одно целое число — количество наборов ответов, удовлетворяющих данному ограничению, по модулю $998244353$.

## Примеры

**входные данные**

```
3 3
1 3 1
```

**выходные данные**

```
9
```

**входные данные**

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

**выходные данные**

```
1000
```

**входные данные**

```
6 2
1 1 2 2 1 1
```

**выходные данные**

```
16
```

### Примечание

На первый пример, корректные наборы ответов — это $[2,1,1],[2,1,2],[2,1,3],[3,1,1],[3,1,2],[3,1,3],[3,2,1],[3,2,2],[3,2,3]$.

## G. Не равны

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Вам задан массив целых чисел $a_1, a_2, \ldots, a_n$, где $a_i$ означает количество блоков на $i$-й позиции. Гарантируется, что $1 \le a_i \le n$.

За одну операцию вы можете выбрать любое подмножество индексов и убрать один блок на каждой позиции в выбранном подмножестве. Вы не можете убрать блок с позиции, на которой уже нет блоков.

Все подмножества которые вы выбираете должны быть различными (уникальными).

Вам необходимо удалить все блоки за не более чем $n+1$ операцию. Можно доказать, что ответ всегда существует.

### Входные данные

В первой строке записано одно целое число $n$ ($1 \le n \le 10^3$) — длина данного массива.

Во второй строке записаны $n$ целых чисел $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) — количество блоков на позициях $1, 2, \ldots, n$.

### Выходные данные

В первой строке выведите целое число $op$ ($0 \le op \le n+1$).

В каждой из следующих $op$ строк выведите бинарную строку $s$ длины $n$.

Если $s_i$='0', это означает, что позиция $i$ не находится в выбранном подмножестве. Иначе, $s_i$ должно быть равно '1', и позиция $i$ находится в выбранном подмножестве.

Все строки должны быть различными и $a_i$ должно быть равно сумме $s_i$ по всем выбранным бинарным строкам.

Если есть несколько возможных ответов, вы можете вывести любой.

Гарантируется, что ответ существует.

### Примеры

входные данные

```
5
5 5 5 5 5
```

выходные данные

```
6
11111
01111
10111
11011
11101
11110
```

входные данные

```
5
5 1 1 1 1
```

выходные данные

```
5
11000
10000
10100
10010
10001
```

| входные данные |
| --- |
| 5<br>4 1 5 3 4 |

| выходные данные |
| --- |
| 5<br>11111<br>10111<br>10101<br>00111<br>10100 |

### Примечание

В первом примере количество блоков уменьшается следующим образом: $\{5,5,5,5,5\} \to \{4,4,4,4,4\} \to \{4,3,3,3,3\} \to \{3,3,2,2,2\} \to \{2,2,2,1,1\} \to \{1,1,1,1,0\} \to \{0,0,0,0,0\}$. Можно заметить, что все эти операции отличаются друг от друга.