B. Егор и граф

ограничение по времени на тест: 3 секунды
ограничение по памяти на тест: 256 мегабайт

У Егора есть взвешенный ориентированный граф, состоящий из n вершин. В этом графе между любой парой различных вершин есть ребро в обоих направлениях. Егор любит играть с графом, и сейчас он придумал новую игру:

- Игра состоит из n шагов.
- На i-том шаге Егор удаляет из графа вершину номер x_i. Удаляя вершину, Егор удаляет все ребра, которые входили в данную вершину и которые выходили из нее.
- Перед выполнением каждого шага, Егор хочет знать сумму длин кратчайших путей между всеми парами оставшихся вершин. Кратчайший путь может проходить через любую оставшуюся вершину. Другими словами, если обозначить как $d(i, v, u)$ кратчайший путь между вершинами v и u в графе, который получился до удаления вершины x_i, то Егор хочет знать значение следующей суммы: $\sum_{v,u,v\neq u} d(i,v,u)$.

Помогите Егору, выведите значение искомой суммы перед каждым шагом.

Входные данные

В первой строке содержится целое число n $(1 \le n \le 500)$ — количество вершин в графе.

В следующих n строках содержится по n целых чисел — матрица смежности графа: j-тое число в i-той строке a_{ij} $(1 \le a_{ij} \le 10^5, a_{ii} = 0)$ обозначает вес ребра, ведущего из вершины i в вершину j.

В следующей строке содержится n различных целых чисел: $x_1, x_2, ..., x_n$ $(1 \le x_i \le n)$ — вершины, которые удаляет Егор.

Выходные данные

Выведите n целых чисел — i-тое число равно искомой сумме перед i-тым шагом.

Пожалуйста, не используйте спецификатор `%lld` для чтения или записи 64-х битовых чисел на *C++*. Рекомендуется использовать потоки `cin`, `cout` или спецификатор `%I64d`.

Примеры

входные данные Скопировать

```
1
0
1
```

выходные данные Скопировать

```
0
```

входные данные Скопировать

```
2
0 5
4 0
1 2
```

выходные данные Скопировать

```
9 0
```

входные данные Скопировать

```
4
0 3 1 1
6 0 400 1
2 4 0 1
1 1 1 0
4 1 2 3
```

выходные данные Скопировать

```
17 23 404 0
```

D. Солдат и игра с числами

ограничение по времени на тест: 3 секунды
ограничение по памяти на тест: 256 мегабайт

Два солдата играют в игру. Сначала один солдат выбирает целое положительное число n и называет его второму солдату. Затем второй солдат пытается совершить наибольшее количество ходов. На каждом ходу солдат выбирает положительное целое число $x > 1$, такое, что n делится на x, и заменяет число n числом n / x. Когда n становится равно 1 и больше нет возможных ходов, тогда игра завершается и счет второго солдата равен количеству совершённых им ходов.

Чтобы сделать игру поинтереснее, в качестве n солдат выбирает число вида $a! / b!$ для некоторых положительных целых чисел a и b ($a \geq b$). Здесь под $k!$ имеется в виду *факториал* числа k, который определяется как произведение всех положительных целых чисел, не превосходящих k.

Определите максимально возможный счет второго солдата.

Входные данные

В первой строке входа записано единственное целое число t ($1 \leq t \leq 1\,000\,000$) обозначающее количество игр, в которые играют солдаты.

Затем следуют t строк, в каждой строке записана пара чисел a и b ($1 \leq b \leq a \leq 5\,000\,000$), обозначающая значение n для игры.

Выходные данные

Для каждой игры выведите максимальный счет, которого может достичь второй солдат.

Примеры

входные данные	Скопировать
2 3 1 6 3	

выходные данные	Скопировать
2 5	

B. Сумма по модулю

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Дана последовательность чисел $a_1, a_2, ..., a_n$, а также число m.

Проверьте, можно ли выбрать непустую подпоследовательность a_{i_j} такую, что сумма чисел в этой подпоследовательности делится на m.

Входные данные

В первой строке даны два числа n и m ($1 \le n \le 10^6$, $2 \le m \le 10^3$) — размер исходной последовательности и число, от деления на которое берётся остаток у суммы.

Во второй строке даны n чисел $a_1, a_2, ..., a_n$ ($0 \le a_i \le 10^9$).

Выходные данные

В единственной строке выведите «YES» (без кавычек) в случае, если существует требуемая подпоследовательность, либо «NO» (без кавычек), если такой подпоследовательности не существует.

Примеры

входные данные	Скопировать
3 5 1 2 3	
выходные данные	**Скопировать**
YES	

входные данные	Скопировать
1 6 5	
выходные данные	**Скопировать**
NO	

входные данные	Скопировать
4 6 3 1 1 3	
выходные данные	**Скопировать**
YES	

входные данные	Скопировать
6 6 5 5 5 5 5 5	
выходные данные	**Скопировать**
YES	

Примечание

В первом тесте из условия можно выбрать числа 2 и 3, сумма которых делится на 5.

Во втором тесте из условия, единственная непустая подпоследовательность чисел — одно число 5. Число 5 не делится на 6, стало быть, искомой подпоследовательности не существует.

В третьем тесте из условия нужно выбрать два числа 3 на концах.

В четвертом тесте из условия можно взять целиком всю последовательность.

B. Zuma

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт

Генос недавно установил на свой телефон игру «Zuma». Игроку даётся ряд из n драгоценных камней, i-й из которых имеет цвет c_i. Цель игры — уничтожить все камни за как можно меньшее количество секунд.

За одну секунду Генос может выбрать любую подстроку (последовательность стоящих рядом камней), являющуюся палиндромом, и удалить её. После удаления данной подстроки оставшиеся камни сдвигаются, чтобы снова образовать непрерывный ряд. Какое минимальное количество секунд необходимо, чтобы уничтожить всю строку?

Напомним, что строка (или подстрока) является *палиндромом*, если она одинаково читается как слева направо, так и справа налево. В данном случае это означает, что цвет первого камня равен цвету последнего камня, цвет второго равен цвету предпоследнего и так далее.

Входные данные

В первой строке входных данных записано единственное число n ($1 \le n \le 500$) — количество камней в изначальном ряду.

Во второй строке записано n целых чисел, i-е из которых равно c_i ($1 \le c_i \le n$) — цвет i-го камня в изначальном ряду.

Выходные данные

Выведите единственное целое число — минимальное количество секунд, необходимое чтобы удалить все камни.

Примеры

входные данные	Скопировать
3 1 2 1	

выходные данные	Скопировать
1	

входные данные	Скопировать
3 1 2 3	

выходные данные	Скопировать
3	

входные данные	Скопировать
7 1 4 4 2 3 2 1	

выходные данные	Скопировать
2	

Примечание

В первом примере Генос может уничтожить весь ряд за 1 секунду.

Во втором примере Генос может уничтожать только по одному камню в секунду за раз, поэтому на весь ряд у него уйдёт 3 секунды.

В третьем примере один из возможных способов достижения оптимального времени — сперва уничтожить $4\ \ 4$, а затем уничтожить $1\ \ 2\ \ 3\ \ 2\ \ 1$.

E. Режим сна

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

У Вовы довольно странный режим сна. В сутках h часов. Вова будет спать ровно n раз. В i-й раз он будет спать ровно после a_i часов после пробуждения. Предположим, что Вова просыпается ровно в начале этой истории (изначальное время равно 0). Каждый раз Вова спит ровно сутки (другими словами, h часов).

Вова думает, что i-й раз, когда он спит, — **хороший**, если он идет спать между l и r часами включительно.

Вова может контролировать себя и перед i-м разом он может выбрать одно из двух: пойти спать после a_i часов или после $a_i - 1$ часов.

Ваша задача — назвать максимальное число **хороших** раз, которые Вова может получить, если будет действовать оптимально.

Входные данные

Первая строка теста содержит четыре целых числа n, h, l и r ($1 \le n \le 2000, 3 \le h \le 2000, 0 \le l \le r < h$) — количество раз, которое Вова будет спать, количество часов в сутках и отрезок **хорошего** времени для сна.

Вторая строка теста содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \le a_i < h$), где a_i — количество часов, после которого Вова пойдет спать в i-й раз.

Выходные данные

Выведите одно целое число — максимальное количество **хороших** раз, которое Вова может получить, если будет действовать оптимально.

Пример

входные данные	Скопировать
7 24 21 23 16 17 14 20 20 11 22	

выходные данные	Скопировать
3	

Примечание

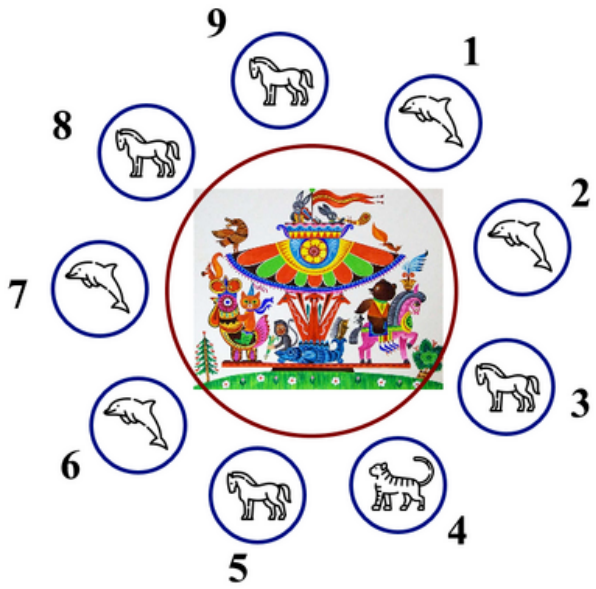
Максимальное количество **хороших** раз в тестовом примере равно 3.

История начинается с $t = 0$. Затем Вова идет спать после $a_1 - 1$ часов, теперь время равно 15. Это время не является хорошим. Затем Вова идет спать после $a_2 - 1$ часов, теперь время равно $15 + 16 = 7$. Это время также не является хорошим. Затем Вова идет спать после a_3 часов, теперь время равно $7 + 14 = 21$. Это время является **хорошим**. Затем Вова идет спать после $a_4 - 1$ часов, теперь время равно $21 + 19 = 16$. Это время не является хорошим. Затем Вова идет спать после a_5 часов, теперь время равно $16 + 20 = 12$. Это время не является хорошим. Затем Вова идет спать после a_6 часов, Теперь время равно $12 + 11 = 23$. Это время является **хорошим**. Затем Вова идет спать после a_7 часов, теперь время равно $23 + 22 = 21$. Это время также является **хорошим**.

D. Карусель, карусель — это радость для нас

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Круглая карусель состоит из n фигур различных животных. Фигуры пронумерованы от 1 до n по ходу движения карусели. Таким образом, после n-й фигуры идёт фигура с номером 1. Каждая фигура имеет вид — это животного этой фигуры (лошадка, тигр и др.). Вид животного i-й фигуры равен t_i.



Пример изображения карусели для $n = 9$ и $t = [5, 5, 1, 15, 1, 5, 5, 1, 1]$.

Вы хотите покрасить каждую фигуру в один из цветов. Вам кажется скучным, если в каруселе разные фигуры (с разными видами животных) идут подряд и покрашены в одинаковые цвета.

Ваша задача — покрасить фигуры так, чтобы количество различных использованных цветов было минимальным и не существовало двух фигур разного вида, которые идут подряд и покрашены в один цвет одновременно. Если Вы используете ровно k различных цветов, то цвета фигур должны быть пронумерованы целыми числами от 1 до k.

Входные данные

Входные данные состоят из одного или более набора входных данных.

В первой строке записано целое число q ($1 \le q \le 10^4$) — количество наборов входных данных в тесте. Далее следуют описания q наборов входных данных, по две строки на каждый набор.

Первая строка набора входных данных содержит одно целое число n ($3 \le n \le 2 \cdot 10^5$) — количество фигур на карусели. Фигуры пронумерованы от 1 до n по ходу вращения карусели. Считайте, что после фигуры n идёт фигура 1.

Вторая строка набора входных данных содержит n целых чисел $t_1, t_2, \ldots, t_n$ ($1 \le t_i \le 2 \cdot 10^5$), где t_i равно виду животного i-й фигуры.

Сумма значений n по всем наборам входных данных в тесте не превосходит $2 \cdot 10^5$.

Выходные данные

Выведите q ответов — на для каждого набора входных данных выведите две строки.

В первую строку выведите одно целое число k — минимально возможное количество различных цветов фигур.

Во вторую строку выведите n целых чисел $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le k$), где c_i обозначает номер цвета i-й фигуры. Если существует несколько возможных ответов, вы можете вывести любой из них.

Пример

входные данные	Скопировать
4 5	

```
1 2 1 2 2
6
1 2 2 1 2 2
5
1 2 1 2 3
3
10 10 10
```

выходные данные Скопировать

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


ИТМО

F. Строка-шпион

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам даны n строк $a_1, a_2, \ldots, a_n$, все они имеют одинаковую длину m. Строки состоят из строчных букв латинского алфавита.

Найдите любую такую строку s длины m, что каждая из заданных n строк отличается от s не более чем в одной позиции. Формально, для каждой заданной строки a_i должно существовать не более одной позиции j, в которой $a_i[j] \neq s[j]$.

Заметим, что искомая строка s может как совпадать с одной из заданных строк a_i, так и отличаться от всех заданных строк.

Например, если вам даны строки abac и zbab, тогда ответом на задачу может быть строка abab, которая отличается от первой только последним символом, а от второй только первым.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 100$) — количество наборов тестовых данных в тесте. Далее записаны t наборов тестовых данных.

Каждый набор начинается со строки, в которой записаны целые числа n ($1 \leq n \leq 10$) и m ($1 \leq m \leq 10$) — количество заданных строк и их длина.

Далее следуют n строк a_i. Каждая имеет длину m и состоит из строчных латинских букв.

Выходные данные

Выведите t ответов на наборы тестовых данных. Каждый ответ (если он существует) — это строка длины m, состоящая из строчных латинских букв. Если существует несколько ответов, то выведите любой из них. Если ответа не существует, выведите «-1» («минус один», без кавычек).

Пример

входные данные Скопировать

```
5
2 4
abac
zbab
2 4
aaaa
bbbb
3 3
baa
aaa
aab
2 2
ab
bb
3 1
a
b
c
```

выходные данные Скопировать

```
abab
-1
aaa
ab
z
```

Примечание

Первый набор тестовых данных примера разобран в условии.

Во втором наборе ответа не существует.

D. Цветные прямоугольники

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Даны три мультимножества пар цветных палок:

- R пар красных палок, у первой пары длины равны r_1, у второй пары длины равны r_2, ..., у R-й пары длины равны r_R;
- G пар зеленых палок, у первой пары длины равны g_1, у второй пары длины равны g_2, ..., у G-й пары длины равны g_G;
- B пар синих палок, у первой пары длины равны b_1, у второй пары длины равны b_2, ..., у B-й пары длины равны b_B.

Вы собираете прямоугольники из этих пар палок следующим образом:

1. взять пару палок одного цвета;
2. взять пару палок другого цвета, отличного от первого;
3. прибавить площадь полученного прямоугольника к суммарной площади.

В итоге получатся такие прямоугольники, что противоположные стороны у них одного цвета, а соседние стороны различных цветов.

Каждая пара палок может быть использована не более одного раза, некоторые пары можно не использовать. Не разрешается разбивать пару палок на отдельные палки.

Какую максимальную суммарную площадь можно получить?

Входные данные

В первой строке записаны три целых числа R, G, B ($1 \le R, G, B \le 200$) — количество пар красных палок, количество пар зеленых палок и количество пар синих палок.

Во второй строке записаны R целых чисел $r_1, r_2, \ldots, r_R$ ($1 \le r_i \le 2000$) — длины палок в каждой паре красных палок.

В третьей строке записаны G целых чисел $g_1, g_2, \ldots, g_G$ ($1 \le g_i \le 2000$) — длины палок в каждой паре зеленых палок.

В четвертой строке записаны B целых чисел $b_1, b_2, \ldots, b_B$ ($1 \le b_i \le 2000$) — длины палок в каждой паре синих палок.

Выходные данные

Выведите максимально возможную суммарную площадь построенных прямоугольников.

Примеры

входные данные Скопировать

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

выходные данные Скопировать

```
20
```

входные данные Скопировать

```
2 1 3
9 5
1
2 8 5
```

выходные данные Скопировать

```
99
```

входные данные Скопировать

```
10 1 1
11 7 20 15 19 14 2 4 13 14
8
11
```

выходные данные Скопировать

```
372
```

Примечание

В первом примере можно построить один из следующих прямоугольников: красный и зеленый со сторонами 3 и 5, красный и синий со сторонами 3 и 4 и зеленый и синий со сторонами 5 и 4. Лучшая площадь из них равна $4 \times 5 = 20$.

Во втором примере лучшие прямоугольники: красный/синий 9×8, красный/синий 5×5, зеленый/синий 2×1. Суммарная площадь равна $72 + 25 + 2 = 99$.

В третьем примере лучшие прямоугольники: красный/зеленый 19×8 и красный/синий 20×11. Суммарная площадь равна $152 + 220 = 372$. Обратите внимание, что больше прямоугольников нельзя построить, потому что не разрешается иметь обе взятые пары одного цвета.

C. Спортивный фестиваль

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Студенческий совет готовится к эстафете на спортивном фестивале.

Совет состоит из n членов. Они будут бегать один за другим в забеге, скорость члена i равна s_i. *Расхождение* d_i i-го этапа — разница между максимальной и минимальной скоростью бега среди первых i членов, которые участвовали в забеге. Формально, если a_i обозначает скорость i-го участника, участвовавшего в забеге, то $d_i = \max(a_1, a_2, \ldots, a_i) - \min(a_1, a_2, \ldots, a_i)$.

Вы хотите минимизировать сумму расхождений $d_1 + d_2 + \cdots + d_n$. Для этого можно изменить порядок, в котором будут бежать участники. Какова минимально возможная сумма?

Входные данные

В первой строке содержится единственное целое число n ($1 \le n \le 2000$) — количество членов студенческого совета.

Вторая строка содержит n целых чисел $s_1, s_2, \ldots, s_n$ ($1 \le s_i \le 10^9$) — скорости членов совета.

Выходные данные

Выведите единственное целое число — минимально возможное значение $d_1 + d_2 + \cdots + d_n$ после выбора порядка, в котором будут бежать члены совета.

Примеры

входные данные	Скопировать
3 3 1 2	
выходные данные	**Скопировать**
3	

входные данные	Скопировать
1 5	
выходные данные	**Скопировать**
0	

входные данные	Скопировать
6 1 6 3 3 6 3	
выходные данные	**Скопировать**
11	

входные данные	Скопировать
6 104 943872923 6589 889921234 1000000000 69	
выходные данные	**Скопировать**
2833800505	

Примечание

В первом примере, мы можем выбрать, чтобы третий член бежал первым, затем первый, и, наконец, второй. Таким образом, $a_1 = 2$, $a_2 = 3$, а $a_3 = 1$. Тогда получаем:

- $d_1 = \max(2) - \min(2) = 2 - 2 = 0$.
- $d_2 = \max(2, 3) - \min(2, 3) = 3 - 2 = 1$.
- $d_3 = \max(2, 3, 1) - \min(2, 3, 1) = 3 - 1 = 2$.

Полученная сумма равна $d_1 + d_2 + d_3 = 0 + 1 + 2 = 3$. Можно показать, что меньшего значения добиться невозможно.

Во втором примере единственная возможная перестановка дает $d_1 = 0$, поэтому минимально возможный результат равен 0.

C. Ехаб снова делит массив

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Ехаб снова решил развлечься. У него есть массив a длины n. Он называет массив хорошим, если его нельзя разбить на 2 непересекающихся подпоследовательности такие, что сумма элементов первой равна сумме элементов второй. Сейчас он хочет удалить минимальное количество элементов из a так, что оставшийся массив будет хорошим. Вы можете ему в этом помочь?

Последовательность a является подпоследовательностью b, если a может быть получена из b удалением нескольких (возможно, ни одного или всех) элементов. Разбить массив это разделить его на 2 подпоследовательности так, чтобы каждый элемент входил ровно в одну из них. Вы должны использовать все элементы по одному разу.

Входные данные
В первой строке записано одно целое число n ($2 \le n \le 100$) — размер массива a.

Во второй строке записаны n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 2000$) — элементы массива a.

Выходные данные
Первая строка должна содержать минимальное количество элементов, которое нужно удалить.

Вторая строка должна содержать индексы этих элементов, записанные через пробел.

Если существует несколько решений, выведите любое из них. Можно показать, что решение всегда существует.

Примеры

входные данные	Скопировать
4 6 3 9 12	

выходные данные	Скопировать
1 2	

входные данные	Скопировать
2 1 2	

выходные данные	Скопировать
0	

Примечание
В первом примере вы можете разбить массив на $[6, 9]$ и $[3, 12]$, так что необходимо удалить хотя бы 1 элемент. Удаление 3 подходит.

Во втором примере массив уже хороший.