

## Задача А. Бутфол

Имя входного файла: `bootfall.in`  
Имя выходного файла: `bootfall.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Тима и его  $N$  друзей очень любят играть в *Бутфол*. *Бутфол* — спортивная игра, в которой участвуют  $N + 1$  игроков. Каждый игрок имеет силу, которая характеризуется целым положительным числом. Игра состоит из  $N + 1$  раундов, в каждом раунде кто-то из игроков записывает раунд на видео, а остальные  $N$  игроков делятся на две команды так, что каждый игрок будет в одной из двух команд и обе команды непустые. Сила команды — это сумма сил всех игроков в команде. Также, каждый игрок должен снимать на видео **ровно в одном** раунде.

Раунд называется *ничейным* если существует разбиение на две команды с **равной** силой, и игра называется *дружной* если **все раунды ничейные**. Каждый из  $N$  друзей сообщил Тиме свою силу, а сам Тима может выбрать себе силу любой допустимой величины.

По заданным значениям сил  $N$  друзей, помогите Тиме определить все варианты сил, которые он может выбрать себе так, чтобы игра могла стать *дружной*.

### Формат входных данных

В первой строке входных данных находится целое число  $N$  ( $1 \leq N \leq 500$ ) — количество друзей у Тимы. Во второй строке находятся  $N$  целых чисел  $a_1, a_2, \dots, a_N$  ( $1 \leq a_i \leq 500$ ;  $1 \leq i \leq N$ ) разделенных через единичный пробел,  $a_i$  — сила  $i$ -ого игрока.

### Формат выходных данных

В первой строке выведите одно целое число  $K$  — количество способов выбрать Тиме силу. Если же не существует вариантов сил для Тимы, выведите «0» (без кавычек), иначе во второй строке выведите  $K$  целых положительных чисел разделенных через одиночный пробел — сами значения сил для Тимы, значения следует выводить в **возрастающем** порядке.

### Система оценки

Данная задача содержит шесть подзадач:

- $1 \leq N \leq 12$ ,  $1 \leq a_i \leq 200$ , для всех  $1 \leq i \leq N$ . Оценивается в 6 баллов.
- $1 \leq N \leq 30$ ,  $1 \leq a_i \leq 20$ , для всех  $1 \leq i \leq N$ . Оценивается в 7 баллов.
- $1 \leq N \leq 100$ ,  $1 \leq a_i \leq 100$ , для всех  $1 \leq i \leq N$ . Оценивается в 15 баллов.
- $1 \leq N \leq 270$ ,  $1 \leq a_i \leq 270$ , для всех  $1 \leq i \leq N$ . Оценивается в 16 баллов.
- $1 \leq N \leq 350$ ,  $1 \leq a_i \leq 350$ , для всех  $1 \leq i \leq N$ . Оценивается в 21 баллов.
- $1 \leq N \leq 500$ ,  $1 \leq a_i \leq 500$ , для всех  $1 \leq i \leq N$ . Оценивается в 35 баллов.

Каждая подзадача оценивается только при прохождении всех предыдущих.

### Примеры

<code>bootfall.in</code>	<code>bootfall.out</code>
4 1 3 1 5	1 3
6 3 5 7 11 9 13	4 1 3 17 19
3 2 2 2	0
4 200 200 200 200	2 200 600

## Замечание

**Пояснение к первому примеру.**

Покажем, что если Тима выберет себе силу 3, то игра может быть *дружной*.

— Когда Тима будет отвечать за съемку, чтобы раунд был *ничейным*, остальные могут поделиться таким образом : (1, 3, 1) в первой команде, и (5) во второй.

— Когда друг с номером 1 будет отвечать за съемку, остальные могут поделиться таким образом: (1, 5) в первой команде, (3, 3) во второй.

— Когда друг с номером 2 будет отвечать за съемку, остальные могут поделиться таким образом: (1, 1, 3) в первой команде, (5) во второй.

— Когда друг с номером 3 будет отвечать за съемку, остальные могут поделиться таким образом: (3, 3) в первой команде, (1, 5) во второй.

— Когда друг с номером 4 будет отвечать за съемку, остальные могут поделиться таким образом: (1, 3) в первой команде, (1, 3) во второй.

Если Тима выберет себе силу не равной 3, то игра не может быть *дружной*.

## Задача В. Банкноты

Имя входного файла: `money.in`  
Имя выходного файла: `money.out`  
Ограничение по времени: 1.5 секунд  
Ограничение по памяти: 256 мегабайт

АланашКО очень любит деньги. В преддверии Нового года ему подарили  $N$  банкнот. Номинал каждой банкноты является целым положительным числом. Играясь, АланашКО разложил все банкноты в ряд и пронумеровал банкноты слева направо от 1 до  $N$ . Затем он решил отсортировать все банкноты в порядке **неубывания**. Для этого АланашКО поступает следующим образом: сперва он делит весь ряд банкнот на один или несколько **непересекающихся** подотрезков так, чтобы **каждая** банкнота находилась в каком-либо подотрезке. Далее все подотрезки в порядке слева направо поочередно вставляются в новый ряд, т.е. сперва вставляется самый левый подотрезок (первый подотрезок), затем следующий самый левый и так далее. Каждый подотрезок целиком вставляется либо между любыми двумя банкнотами, либо в один из двух концов текущего ряда. Порядок банкнот внутри подотрезка не изменяется при вставке.

АланашКО хочет минимизировать количество подотрезков так, чтобы он смог в итоге отсортировать банкноты в порядке **неубывания** номиналов. Помогите ему найти это значение.

### Формат входных данных

В первой строке входных данных дается целое положительное число  $N$  ( $1 \leq N \leq 10^6$ ) — количество банкнот. Во второй строке входных данных дается  $N$  целых положительных чисел  $a_i$  ( $1 \leq a_i \leq 10^6$ ) — номинал  $i$ -ой слева банкноты в изначальном ряде.

### Формат выходных данных

В единственной строке выходных данных выведите одно число — минимальное количество подотрезков, при котором АланашКО способен отсортировать ряд.

### Система оценки

Данная задача содержит четыре подзадачи:

1.  $N \leq 8$ . Оценивается в 9 баллов.
2.  $N \leq 20$ . Оценивается в 16 баллов.
3.  $N \leq 300$ . Оценивается в 20 баллов.
4.  $N \leq 10^6$ . Оценивается в 55 баллов.

Каждая подзадача оценивается только при прохождении всех предыдущих.

### Пример

<code>money.in</code>	<code>money.out</code>
6 3 6 4 5 1 2	3

### Замечание

*Подотрезком* — называется некая подряд идущая последовательность.

Рассмотрим тест из условия:

Минимальным ответом будет разбиение массива на 3 подотрезка:  $|3\ 6|4\ 5|1\ 2|$  (палочки — границы подотрезков)

После первого хода: изначальный ряд  $|4\ 5|1\ 2|$ , текущий ряд  $|3\ 6|$ .

На втором ходу подотрезок  $|4\ 5|$  может встать между 3 и 6.

После второго хода: изначальный ряд  $|1\ 2|$ , текущий ряд:  $|3\ 4\ 5\ 6|$ . Затем, подотрезок  $|1\ 2|$  вставляется в начало текущего ряда и получается  $|1\ 2\ 3\ 4\ 5\ 6|$ .

## Задача С. Наидлиннейшая красивая последовательность

Имя входного файла: `subsequence.in`  
Имя выходного файла: `subsequence.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Вам даны две последовательности целых неотрицательных чисел размера  $n$ :  $a_1, a_2, \dots, a_n$  и  $k_1, k_2, \dots, k_n$ . Последовательность из  $m$  целых чисел  $i_1, i_2, \dots, i_m$  будет называться *красивой* если выполняется каждое из следующих условий:

- $1 \leq i_1 < i_2 < \dots < i_m \leq n$ . Иными словами, последовательность должна быть возрастающей.
- $\text{bitCount}(a_{i_{j-1}} \text{ AND } a_{i_j}) = k_{i_j}$  для всех  $1 < j \leq m$ .

Найдите *красивую* последовательность максимальной длины.

### Формат входных данных

В первой строке входных данных дано целое положительное число  $n$  ( $1 \leq n \leq 10^5$ ) — размер последовательности  $a$  и  $k$ . Вторая строка содержит  $n$  целых неотрицательных чисел  $a_i$  ( $0 \leq a_i < 2^{20}$ ) — последовательность  $a$ . В третьей строке содержится  $n$  целых неотрицательных чисел  $k_i$  ( $0 \leq k_i \leq 20$ ) — последовательность  $k$ . Числа в обеих последовательностях задаются через одиночный пробел.

### Формат выходных данных

В первой строке выходных данных выведите целое число  $m$  — размер максимальной *красивой* последовательности. Во второй строке выведите  $m$  чисел — значения максимальной *красивой* последовательности. Если ответов несколько, выведите любой.

### Система оценки

Данная задача содержит четыре подзадачи:

1.  $1 \leq n \leq 15$ ,  $0 \leq a_i < 2^{20}$ . Оценивается в 7 баллов.
2.  $1 \leq n \leq 5000$ ,  $0 \leq a_i < 2^{20}$ . Оценивается в 16 баллов.
3.  $1 \leq n \leq 10^5$ ,  $0 \leq a_i < 2^8$ . Оценивается в 17 баллов.
4.  $1 \leq n \leq 10^5$ ,  $0 \leq a_i < 2^{20}$ . Оценивается в 60 баллов.

Каждая подзадача оценивается только при прохождении всех предыдущих.

### Примеры

subsequence.in	subsequence.out
4 1 2 3 4 10 0 1 0	4 1 2 3 4
2 8 9 20 0	1 1
5 5 3 5 3 5 10 1 20 1 20	2 1 2

### Замечание

$\text{bitCount}(x)$  — это количество единичных битов в двоичном представлении, например:  $\text{bitCount}(5_{10}) = \text{bitCount}(101_2) = 2$ ,  $\text{bitCount}(0) = 0$ ,  $\text{bitCount}(8) = 1$ .

AND — это бинарная операция, действие которой эквивалентно применению логического «И» к каждой паре битов, например:  $11_{10} \text{ AND } 13_{10} = 1011_2 \text{ AND } 1101_2 = 1001_2 = 9$ ,  $7_{10} \text{ AND } 16_{10} = 111_2 \text{ AND } 10000_2 = 0_2 = 0_{10}$ .