

## 1. Коробки (все классы)



Ограничения: время – 1s/Java 2s, память – 32MiB [Послать решение](#) [Посылки](#) Темы Где Обсудить (0)

Коту Бублику очень нравятся коробки. И чем больше коробка, тем лучше. Но коробки могут иметь разные размеры, и выбор коробки, имеющей больший объем, для Бублика является сложной математической задачей.



Напишите программу, которая по размерам двух коробок, имеющих форму параллелепипеда, определяет, какая из коробок имеет больший объем.

В первой строке содержатся три целых числа в диапазоне от 1 до 100 – длина, ширина и высота первой коробки. Во второй строке содержатся три целых числа в диапазоне от 1 до 100 – длина, ширина и высота второй коробки.

Вывести сообщение `FIRST`, если объем первой коробки больше, чем объем второй коробки. Вывести сообщение `SECOND`, если вторая коробка имеет объем больше, чем первая. В случае равенства объемов коробок вывести сообщение `EQUAL`.

Пример ввода

```
5 6 4
7 3 5
```

Пример вывода

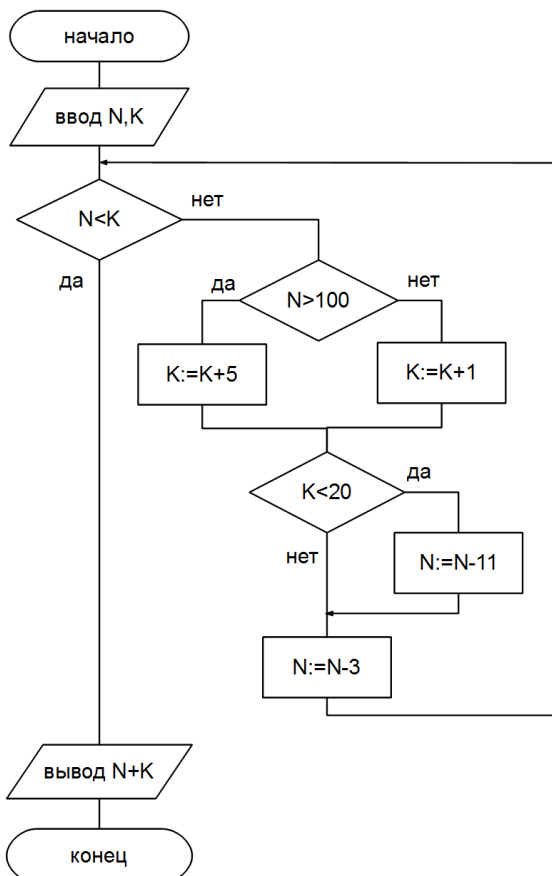
```
FIRST
```

Picture taken from the site [Places You Find Cats](#)

## 2. Алгоритм (8 класс)



Ограничения: время – 1s/Java 2s, память – 32MiB [Послать решение](#) [Посылки](#) Темы Где Обсудить (0)



Реализуйте на одном из языков программирования алгоритм, представленный на схеме.

В первой строке ввода содержатся два целых числа  $N$  и  $K$  ( $1 \leq N, K \leq 1000$ ).

Вывести одно целое число – сумму  $N$  и  $K$  после завершения работы алгоритма.

Пример ввода

```
30 15
```

Пример вывода

```
19
```

## 2. Шифр (9-11 классы)



Ограничения: время – 1s/Java 2s, память – 32MiB [Послать решение](#) [Посылки](#) Темы Где Обсудить (0)

Для открытия замка сейфа в компьютерной игре из серии "Побег из

комнаты" нужно найти шифр. Имеется подсказка – линейка с нанесенными на ней числами, по которой движется ползунок с прорезями, через которые видны числа на линейке. Края ползунка не могут выйти за границы линейки из-за ограничителей на концах линейки. Рядом с прорезями на ползунке нарисованы знаки + и есть текст "max". Нетрудно догадаться, что шифром для сейфа является набор чисел на линейке, видимых через прорези ползунка, сумма которых максимальна.



Напишите программу, которая определит шифр для открытия сейфа.

Первая строка ввода содержит одно целое число  $N$  ( $2 \leq N \leq 10$ ) – размеры ползунка. Вторая строка содержит  $N$  целых чисел от 0 до 1, разделенных пробелами – описание формы ползунка слева направо. Число 0 означает, что эта часть ползунка не имеет прорезей, а число 1 соответствует части ползунка с прорезью. В описании ползунка есть по крайней мере одно число 1. Третья строка ввода содержит одно целое число  $M$  ( $N < M \leq 1000$ ) – количество чисел на линейке. Четвертая строка ввода содержит  $M$  целых чисел от 0 до 99, разделенных пробелами – числа на линейке, перечисленные в порядке их размещения на линейке.

Вывести в первой строке  $K$  целых чисел, разделяя их пробелами – шифр для открытия сейфа, где  $K$  – количество прорезей на ползунке. Числа вывести в том порядке, в котором они видны в прорезях ползунка. Если существует несколько вариантов с максимальной суммой, то вывести вариант, при котором ползунок расположен левее. Допускается вывод лишних ведущих и завершающих пробелов в строке.

Пример ввода

```
5
1 0 0 1 0
10
7 11 4 3 5 15 2 1 0 25
```

Пример вывода

```
4 15
```

Пояснение к примеру: Числа, которые появляются в прорезях при движении ползунка от самой левой возможной позиции до самой правой: 7+3, 11+5, 4+15, 3+2, 5+1, 15+0. Число 25 не может появиться в прорези ползунка из ограничителей на концах линейки. Максимум суммы чисел в прорезях достигается на числах 4 и 15.

### 3. Пиццерия (8 класс)



Ограничения: время – 1s/Java 2s, память – 32MiB [Послать решение](#) [Посылки](#) [Темы](#) [Где](#) [Обсудить](#) (0)

Джон решил построить пиццерию, в которой можно заказать пиццу с доставкой на дом. Пицца будет продаваться по фиксированной цене, и клиент не платит за доставку. Поэтому, если клиент живет слишком далеко от пиццерии, расходы Джона на доставку могут превысить потенциальную прибыль, заложенную в стоимость пиццы. Расходы на доставку зависят от расстояния между пиццерией и домом клиента и не зависят от количества заказанных пицц.

Чем больше пицц заказывает клиент, тем больше прибыль Джона. Джон решил не обслуживать клиентов, для которых расходы на доставку превышают прибыль – они должны заказывать пиццу в другом месте.

На улице, выбранной для строительства пиццерии, расположено  $N$  домов в один ряд. Расстояние между соседними домами будем считать равным одной единице. Киоск-пиццерия будет построен на улице рядом с одним из домов. Предварительно Джон провел опрос и выяснил сколько пицц в день будут покупать в каждом доме. Используя эти данные, Джон хочет найти место для строительства пиццерии, в котором прибыль от продаж будет максимальна. Прибыль Джона рассчитывается как сумма разностей между количеством заказанных в доме пицц и расстоянием от пиццерии до этого дома только для тех домов, где эта разность положительна.

Напишите программу, которая вычисляет максимальную прибыль Джона и расположение пиццерии, обеспечивающее такую прибыль.

В первой строке ввода содержится одно целое число  $N$  ( $1 \leq N \leq 100$ ) – количество домов на улице. Вторая строка ввода содержит  $N$  целых чисел в диапазоне от 0 до 100, разделенных пробелами – информация о количестве пицц, заказываемых в каждом доме.

Вывести в первой строке два целых числа – максимальную прибыль и номер дома, строительство пиццерии рядом с которым обеспечивает рассчитанную максимальную прибыль. Если есть несколько вариантов, обеспечивающих максимальную прибыль, то вывести вариант с наименьшим номером дома.

Пример ввода

```
6
3 1 0 5 0 10
```

Пример вывода

```
13 4
```

Пояснение к примеру: Расстояния до домов от пиццерии возле 4-го дома равны соответственно 3 2 1 0 1 2. Разности между количеством заказанных пицц и расстоянием до пиццерии равны соответственно (3-3) (1-2) (0-1) (5-0) (0-1) (10-2). Положительными являются только разности (5-0) и (10-2), значит прибыль Джона равна (5-0)+(10-2)=13. Такой же результат получается при строительстве пиццерии напротив 5-го и 6-го дома, но по условию задачи нужно вывести наименьший номер. Строительство пиццерии напротив 1-го дома дает прибыль 10, а для 2-го и 3-го дома – 12.

### 3. Пиццерия-2 (9-11 классы)



**Ограничения:** время – 1s/Java 2s, память – 32MiB [Послать решение](#) [Посылки](#) [Темы](#) [Где](#) [Обсудить](#) (0)

Джон решил построить пиццерию, в которой можно заказать пиццу с доставкой на дом. Пицца будет продаваться по фиксированной цене, и клиент не платит за доставку. Поэтому, если клиент живет слишком далеко от пиццерии, расходы Джона на доставку могут превысить потенциальную прибыль, заложенную в стоимость пиццы. Расходы на доставку зависят от расстояния между пиццерией и домом клиента и не зависят от количества заказанных пицц. Чем больше пицц заказывает клиент, тем больше прибыль Джона. Джон решил не обслуживать клиентов, для которых расходы на доставку превышают прибыль – они должны заказывать пиццу в другом месте. Перед строительством пиццерии Джон провел опрос и выяснил сколько пицц в день будут покупать в каждом доме. Используя эти данные, Джон хочет найти место для строительства пиццерии, в котором прибыль от продаж будет максимальна.

Карта города задана с помощью матрицы размером  $N \times M$ . Соседними считаются ячейки с общей стороной. Ячейки матрицы с числом 0 соответствует дорогам, улицам. Можно построить киоск-пиццерию в любой ячейке с 0 и он не будет мешать проезду или проходу. Можно перемещаться из ячейки с 0 только в соседнюю ячейку с 0, и такое перемещение соответствует одной единице расстояния. В любую ячейку с 0 можно попасть, начав движение из любой ячейки с 0. Ячейки матрицы с положительными числами означают дома, в которых будут заказывать пиццу, и число равно количеству заказываемых пицц. У каждой ячейки с положительным числом есть хотя бы одна соседняя ячейка, содержащая число 0. Пицца должна быть доставлена до "двери дома" – в любую соседнюю с домом ячейку с 0. Ячейки с положительными числами не являются проходными. Кроме того, в матрице есть ячейки, содержащие число  $-1$  и соответствующие домам, в которых не будут заказывать пиццу, или каким-то препятствиям для проезда.

Прибыль Джона рассчитывается как сумма разностей между количеством заказанных в доме пицц и минимальным расстоянием по ячейкам с 0 от пиццерии до двери этого дома только для тех домов, где эта разность положительна.

Напишите программу, которая вычисляет максимальную прибыль Джона и расположение пиццерии, обеспечивающее такую прибыль.

В первой строке ввода содержатся два целых числа  $N$  и  $M$  ( $2 \leq N, M \leq 40$ ) – размеры карты города. Далее следует  $N$  строк, содержащих по  $M$  целых чисел в диапазоне от  $-1$  до  $100$ , разделенных пробелами. Существует хотя бы одна ячейка с 0 и хотя бы одна ячейка с положительным числом.

Вывести в первой строке одно целое число – максимальную прибыль. Во второй строке вывести два целых числа, разделенных пробелом – номер строки и номер столбца ячейки, строительство пиццерии в которой обеспечивает рассчитанную максимальную прибыль. Если есть несколько вариантов для ячейки, обеспечивающие максимальную прибыль, то можно вывести любой из них.

Пример ввода

```
3 5
4 0 5 0 10
0 0 2 0 3
6 0 0 0 0
```

Пример вывода

```
21
2 4
```

Пояснение к примеру: Дома с 2 и 3 являются соседними с ячейкой с 0, выбранной для строительства пиццерии, поэтому расстояние от пиццерии до этих домов равно 0. До дверей домов с 5 и 10 расстояние равно 1. До дома с 6 расстояние равно 3, а до дома с 4 – 5. Прибыль Джона равна  $(5-1)+(10-1)+(2-0)+(3-0)+(6-3)=21$ . Дом с 4 в сумму не входит, так как разность  $4-5$  меньше 0.

#### 4. Кругосветное путешествие (8 класс)



**Ограничения:** время – 1s/Java 2s, память – 32MiB [Послать решение](#) [Посылки](#) [Темы](#) [Где Обсудить](#) (0)

Необходимо совершить путешествие по кольцевой автодороге, проходящей через  $N$  городов. Разрешено ехать из  $i$ -го города в  $(i+1)$ -й, а из  $N$ -го города в 1-й. В конце путешествия автомобиль должен вернуться в начальный город. В каждом городе есть заправочная станция, в которой автомобиль ожидает некоторое количество топлива. Также известно количество топлива, необходимое для

переезда в следующий по маршруту город. При круговом путешествии топливо не должно заканчиваться на дороге между городами, но может закончиться, когда автомобиль приезжает на очередную заправочную станцию. В начале путешествия автомобиль стоит на заправочной станции в одном из городов, а бак автомобиля пуст. Бак автомобиля в этой задаче считается безразмерным и способен вместить любое количество топлива. Суммарное количество топлива на заправках в точности равно суммарному количеству топлива, необходимому для кругового путешествия, поэтому начать успешное круговое путешествие возможно не из каждого города.

Напишите программу, которая определяет из каких городов можно начинать, чтобы совершить успешное круговое путешествие.

Первая строка ввода содержит одно целое число  $N$  ( $2 \leq N \leq 100\,000$ ) — количество городов. Вторая строка ввода содержит  $N$  положительных целых чисел, разделенных пробелами — запасы топлива в  $i$ -м городе. Третья строка ввода содержит  $N$  положительных целых чисел, разделенных пробелами — количество топлива, необходимое для переезда из  $i$ -го города в  $(i+1)$ -й, а для  $N$ -го — в 1-й город. Гарантируется, что сумма чисел во второй строке равна сумме чисел в третьей строке и не превосходит  $2^{31}-1$  (максимального значения для `longint`).

Вывести номера городов, подходящих для начала путешествия, в порядке возрастания. Номера городов должны быть разделены пробелом. Допускается вывод с ведущими или завершающими пробелами. Гарантируется, что есть по меньшей мере один подходящий город.

Пример ввода

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

Пример вывода

```
2 3
```

В 50% тестов для этой задачи  $N \leq 1000$ .

#### 4. Разбиение на палиндромы (9-11 классы)



**Ограничения:** время — 1s/Java 2s, память — 32MiB [Послать решение](#) [Посылки](#) [Темы](#) [Где](#) [Обсудить](#) (0)

Палиндромом называется строка, читаемая одинаково слева направо и справа налево. Например, `abba` — палиндром, а `bat` — нет.

Напишите программу, которая подсчитает количество способов разбить заданную строку на непустые подстроки, являющиеся палиндромами. Например, для слова `abbat` есть три способа разбиения на палиндромы: `a-b-b-a-t`, `a-bb-a-t` и `abba-t`.

Первая строка ввода содержит строку из строчных латинских букв длиной от 1 до 100 символов.

Вывести одно целое число — остаток от деления количества способов разбиения на 1 000 000 009.

Пример ввода

```
abbat
```

Пример вывода

```
3
```