

To print higher-resolution math symbols, click the  
**Hi-Res Fonts for Printing** button on the jsMath control panel.

# Задачи

Сайт: [Дистанционная подготовка](#)  
Курс: Д. П. Кириенко. Программирование на языке Python (школа 179 г. Москвы)  
Условия задач: Задачи  
Printed by: maung myo  
Date: Воскресенье 4 Март 2018, 01:25

## Список задач

---

- [Задача А. Возрастает ли список?](#)
- [Задача В. Последний максимум](#)
- [Задача С. Количество цифр](#)
- [Задача D. Противоположные элементы в массиве](#)
- [Задача Е. Сортировка выбором](#)
- [Задача F. Сортировка вставкой](#)
- [Задача G. Сортировка пузырьком](#)
- [Задача H. Сортировка подсчетом](#)
- [Задача I. Создание архива](#)
- [Задача J. Обувной магазин](#)
- [Задача K. Результаты олимпиады](#)
- [Задача L. Такси](#)
- [Задача M. Коньки](#)
- [Задача N. Чемпионат по стрельбе](#)
- [Задача O. SMS-голосование](#)
- [Задача P. Клавиатура](#)
- [Задача Q. Задача Иосифа Флавия](#)
- [Задача R. Умножение многочленов](#)
- [Задача S. Слияние списков](#)
- [Задача T. Пересечение множеств](#)
- [Задача U. Решето Эратосфена](#)
- [Задача V. Шарики](#)
- [Задача W. Колесо Фортуны](#)

- [Задача X. Наибольшее произведение двух чисел](#)
  - [Задача Y. Наибольшее произведение трех чисел](#)
  - [Задача Z. Ханойская сортировка](#)
-

# Возрастает ли список?

## Задача А. Возрастает ли список?

---

Дан список. Определите, является ли он монотонно возрастающим(то есть верно ли, что каждый элемент этого списка больше предыдущего).

Выведите YES, если массив монотонно возрастает и NO в противном случае.

Решение оформите в виде функции `IsAscending(A)`. В данной функции должен быть один цикл `while`, не содержащий вложенных условий и циклов — используйте схему линейного поиска.

Примеры

Входные данные

1 7 9

Выходные данные

YES

---

# Последний максимум

## Задача В. Последний максимум

---

Найдите наибольшее значение в списке и индекс последнего элемента, который имеет данное значение за один проход по списку, не модифицируя этот список и не используя дополнительного списка.

Выведите два значения.

Примеры

Входные данные

1 2 1 2 1

Выходные данные

2 3

---

# Количество цифр

## Задача С. Количество цифр

---

Дана последовательность чисел, состоящая только из чисел 1, ..., 9. Последовательность завершается числом 0. Каждое число записано в отдельной строке.

Подсчитайте, сколько раз в этой последовательности встречаются значения 1, 2, ..., 9. Сохранять всю последовательность введенных чисел в списке нельзя.

Программа должна вывести ровно 9 чисел: количество единиц, двоек, ..., девяток в данной последовательности.

Примеры

Входные данные

4  
9  
3  
4  
2  
4  
9  
4  
3  
8  
0

Выходные данные

0 1 2 4 0 0 1 2

---

# Противоположные элементы в массиве

## Задача D. Противоположные элементы в массиве

---

Дан список чисел. Определите, есть ли в нем два противоположных (то есть дающих в сумме 0) числа. Если такие числа есть в массиве, выведите их индексы в порядке возрастания. Если таких чисел в массиве нет, ничего не выводите. Гарантируется, что таких пар не больше одной.

Примеры

Входные данные

1 2 3 -2 -4

Выходные данные

1 3

---

# Сортировка выбором

## Задача Е. Сортировка выбором

---

Дан список целых чисел. Выведите все элементы этого списка в порядке невозрастания значений. Выведите новый список на экран.

Решите эту задачу при помощи алгоритма сортировки выбором. Решение оформите в виде функции `SelectionSort(A)`.

В алгоритме сортировки выбором мы находим наибольший элемент в списке и ставим его на первое место, затем находим наибольший элемент из оставшихся и ставим его на второе место и т.д.

Примеры

Входные данные

1 4 2 3 4

Выходные данные

4 4 3 2 1

---

# Сортировка вставкой

## Задача F. Сортировка вставкой

---

Дан список целых чисел. Отсортируйте его в порядке неубывания значений. Выведите полученный список на экран.

Решите эту задачу при помощи алгоритма сортировки вставкой. Решение оформите в виде функции `InsertionSort(A)`.

В этой задаче нельзя пользоваться дополнительным списком операциями удаления и вставки элементов.

В алгоритме сортировки вставкой в каждый произвольный момент начальная часть массива уже отсортирована. В решении имеется цикл `for i in range(1, len(A))`, внутри которого предполагается, что элементы списка `A[0], A[1], ..., A[i-1]` уже отсортированы, элемент `A[i]` добавляется в отсортированную часть списка. Для этого находится позиция, в которую необходимо вставить элемент `A[i]`, затем осуществляется циклический сдвиг фрагмента уже отсортированной части.

Примеры

Входные данные

1 4 2 3 4

Выходные данные

1 2 3 4 4

---



# Сортировка пузырьком

## Задача G. Сортировка пузырьком

---

Дан список целых чисел. Отсортируйте его в порядке невозрастания значений. Выведите полученный список на экран.

Решите эту задачу при помощи алгоритма пузырьковой сортировки. Решение оформите в виде функции `BubbleSort(A)`.

В алгоритме пузырьковой сортировки осуществляется проход по списку от начала к концу, и если два соседних элемента списка стоят в неверном порядке, то они переставляются в правильном порядке. В результате минимальный элемент массива окажется на последнем месте. Повторим эту процедуру еще несколько раз, чтобы поставить все элементы на свои места.

Вспомогательным списком пользоваться нельзя.

Примеры

Входные данные

1 4 2 3 4

Выходные данные

4 4 3 2 1

---

# Сортировка подсчетом

## Задача Н. Сортировка подсчетом

---

Дан список из  $N$  ( $N \leq 2 * 10^5$ ) элементов, которые принимают целые значения от 0 до 100.

Отсортируйте этот список в порядке неубывания элементов. Выведите полученный список.

Решение оформите в виде функции CountSort(A), которая модифицирует передаваемый ей список. Использовать встроенные функции сортировки нельзя.

Примеры

Входные данные

7 3 4 2 5

Выходные данные

2 3 4 5 7

---

# Создание архива

## Задача I. Создание архива

---

Системный администратор вспомнил, что давно не делал архива пользовательских файлов. Однако, объем диска, куда он может поместить архив, может быть меньше чем суммарный объем архивируемых файлов.

Известно, какой объем занимают файлы каждого пользователя.

Напишите программу, которая по заданной информации о пользователях и свободному объему на архивном диске определит максимальное число пользователей, чьи данные можно поместить в архив, при этом используя свободное место как можно более полно.

Входные данные

Программа получает на вход в одной строке число  $S$  – размер свободного места на диске (натуральное, не превышает 10000), и число  $N$  – количество пользователей (натуральное, не превышает 100), после этого идет  $N$  чисел - объем данных каждого пользователя (натуральное, не превышает 1000), записанных каждое в отдельной строке.

Выходные данные

Выведите наибольшее количество пользователей, чьи данные могут быть помещены в архив.

Примеры

Входные данные

100 2  
200  
50

Выходные данные

1

Входные данные

100 3  
50  
30  
50

Выходные данные

2

---

# Обувной магазин

## Задача J. Обувной магазин

---

В обувном магазине продается обувь разного размера. Известно, что одну пару обуви можно надеть на другую, если она хотя бы на три размера больше. В магазин пришел покупатель. Требуется определить, какое наибольшее количество пар обуви сможет предложить ему продавец так, чтобы он смог надеть их все одновременно.

Входные данные

Сначала вводится размер ноги покупателя (обувь меньшего размера он надеть не сможет), в следующей строке — размеры каждой пары обуви в магазине через пробел. Размер — натуральное число, не превосходящее 100.

Выходные данные

Выведите единственное число — максимальное количество пар обуви, которое сможет надеть покупатель.

Примеры

Входные данные

60

60 63

Выходные данные

2

Входные данные

26

30 35 40 41 42

Выходные данные

3

---

# Результаты олимпиады

## Задача К. Результаты олимпиады

---

В олимпиаде участвовало  $N$  человек. Каждый получил определенное количество баллов, при этом оказалось, что у всех участников — разное число баллов.

Упорядочите список участников олимпиады в порядке убывания набранных баллов.

Входные данные

Программа получает на вход число участников олимпиады  $N$ . Далее идет  $N$  строк, в каждой строке записана фамилия участника, затем, через пробел, набранное им количество баллов.

Выходные данные

Выведите список участников (только фамилии) в порядке убывания набранных баллов.

Примеры

Входные данные

3

Ivanov 15

Petrov 10

Sidorov 20

Выходные данные

Sidorov

Ivanov

Petrov

---

# Такси

## Задача L. Такси

---

После затянувшегося совещания директор фирмы решил заказать такси, чтобы развезти сотрудников по домам. Он заказал  $N$  машин — ровно столько, сколь у него сотрудников. Однако когда они подъехали, оказалось, что у каждого водителя такси свой тариф за 1 километр.

Директор знает, какому сотруднику сколько километров от работы до дома (к сожалению, все сотрудники живут в разных направлениях, поэтому нельзя отправить двух сотрудников на одной машине). Теперь директор хочет определить, сколько придется заплатить за перевозку всех сотрудников. Естественно, директор хочет заплатить как можно меньшую сумму.

Входные данные

В первой строке записаны  $N$  чисел через пробел, задающих расстояния в километрах от работы до домов сотрудников компании. Во второй строке записаны  $N$  чисел — тарифы за проезд одного километра в такси.

Выходные данные

Выведите одно целое число — наименьшую сумму, которую придется заплатить за доставку всех сотрудников.

Примеры

Входные данные

10 20 30

50 20 30

Выходные данные

1700

---

# Коньки

## Задача М. Коньки

---

В 179 школе есть много коньков самых разных размеров. Школьник может надеть коньки любого размера, который не меньше размеров его ноги. Известны размеры всех коньков и размеры ног школьников. Определите, какое наибольшее число школьников сможет одновременно пойти покататься.

Входные данные

Первая строка содержит список натуральных чисел размеры коньков. Вторая строка содержит список размеров ног школьников.

Выходные данные

Выведите единственное число — наибольшее количество школьников, которое сможет пойти на каток.

Примеры

Входные данные

41 40 39 42

42 41 42

Выходные данные

2

---

# Чемпионат по стрельбе

## Задача N. Чемпионат по стрельбе

---

На региональном этапе Всероссийской олимпиады школьников по информатике 23 января 2011 года предлагалась следующая задача.

Победитель школьного этапа олимпиады по информатике нашел дома в старых бумагах результаты чемпионата страны по стрельбе из лука, в котором участвовал его папа. К сожалению, листок с результатами сильно пострадал от времени, и разобрать фамилии участников было невозможно. Остались только набранные каждым участником очки, причем расположились они в том порядке, в котором участники чемпионата выполняли стрельбу.

Расспросив папу, школьник выяснил, что количество очков, которое набрал папа, заканчивается на **5**, один из победителей чемпионата стрелял раньше папы, а папин друг, который стрелял сразу после папы, набрал меньше очков. Теперь он заинтересовался, какое самое высокое место мог занять его папа на том чемпионате.

Будем считать, что участник соревнования занял  $k$ -е место, если ровно  $k-1$  участников чемпионата набрали строго больше очков, чем он. При этом победителями считались все участники чемпионата, занявшие первое место.

Требуется написать программу, которая по заданным результатам чемпионата определяет, какое самое высокое место на чемпионате мог занять папа победителя школьного этапа олимпиады по информатике.

Входные данные

Программа получает на вход целое число  $n$  — количество участников чемпионата страны по стрельбе ( $3 \leq n \leq 10^5$ ). Далее идет  $n$  положительных целых чисел, каждое из которых не превышает **1000**, — очки участников чемпионата, приведенные в том порядке, в котором они выполняли стрельбу.

Выходные данные

Выведите одно целое число — самое высокое место, которое мог занять папа школьника. Если не существует ни одного участника чемпионата, который удовлетворяет описанным выше условиям, выведите в выходной файл число **0**.

Примеры

Входные данные

7

10 20 15 10 30 5 1

Выходные данные

6



Входные данные

3

15 15 10

Выходные данные

1

Входные данные

3

10 15 20

Выходные данные

0

---

# SMS-голосование

## Задача O. SMS-голосование

---

На телевизионном шоу зрители голосуют за участников шоу, отправляя SMS-сообщение с номером участника. Определите победителя шоу на основе присланных SMS-сообщений.

Входные данные

Первая строка входных данных содержит количество участников шоу  $N \leq 10^6$ . Во второй строке содержится список номеров, присланных телезрителями, через пробел.

Выходные данные

Программа должна вывести номер участника, набравшего наибольшее число голосов. Если таких несколько, то программа должна вывести номера всех победителей шоу в порядке возрастания.

Неверные номера, присланные телезрителями (то есть большие, чем  $N$ ) необходимо игнорировать.

Примеры

Входные данные

3

1 2 2 1 3 2 3

Выходные данные

2

Входные данные

2

1 2 2 1

Выходные данные

1 2

Входные данные

3

4 3 4

Выходные данные

3

---

# Клавиатура

## Задача Р. Клавиатура

---

На региональном этапе Всероссийской олимпиады школьников по информатике в 2009 году предлагалась следующая задача.

Всем известно, что со временем клавиатура изнашивается, и клавиши на ней начинают залипать. Конечно, некоторое время такую клавиатуру еще можно использовать, но для нажатий клавиш приходится использовать большую силу.

При изготовлении клавиатуры изначально для каждой клавиши задается количество нажатий, которое она должна выдерживать. Если знать эти величины для используемой клавиатуры, то для определенной последовательности нажатых клавиш можно определить, какие клавиши в процессе их использования сломаются, а какие — нет.

Требуется написать программу, определяющую, какие клавиши сломаются в процессе заданного варианта эксплуатации клавиатуры.

Входные данные

Первая строка входных данных содержит целое число  $n$  ( $1 \leq n \leq 1000$ ) — количество клавиш на клавиатуре. Вторая строка содержит  $n$  целых чисел —  $c_1, c_2, \dots, c_n$ , где  $c_i$  ( $1 \leq c_i \leq 100000$ ) — количество нажатий, выдерживаемых  $i$ -ой клавишей. Третья строка содержит целое число  $k$  ( $1 \leq k \leq 100000$ ) — общее количество нажатий клавиш, и последняя строка содержит  $k$  целых чисел  $p_j$  ( $1 \leq p_j \leq n$ ) — последовательность нажатых клавиш.

Выходные данные

Программа должна вывести  $n$  строк, содержащих информацию об исправности клавиш. Если  $i$ -я клавиша сломалась, то  $i$ -ая строка должна содержать слово YES, если же клавиша работоспособна — слово NO.

Примеры

Входные данные

```
5
1 50 3 4 3
16
1 2 3 4 5 1 3 3 4 5 5 5 5 4 5
```

Выходные данные

```
YES
NO
NO
NO
YES
```



# Задача Иосифа Флавия

## Задача Q. Задача Иосифа Флавия

---

$N$  человек, пронумерованных числами от 1 до  $N$  стоят в кругу. Они начинают считаться, каждый  $K$ -й по счету человек выбывает из круга, после чего счет продолжается со следующего за ним человека.

Определите номер человека, который останется в кругу последним.

Входные данные

Программа получает на вход числа  $N$  и  $K$ , не превосходящие 100

Выходные данные

Программа должна вывести одно число от 1 до  $N$ , являющееся ответом на поставленную задачу.

Пояснение к примеру

В этом примере люди выбывают в таком порядке: 2, 5, 1, 3, остался номер 4.

Примеры

Входные данные

5 7

Выходные данные

4

---

# Умножение многочленов

## Задача R. Умножение многочленов

---

Даны коэффициенты двух многочленов:  $(A(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n)$  и  $(B(x) = b_0 + b_1x + b_2x^2 + \dots + b_mx^m)$ .

Найдите коэффициенты многочлена, являющегося их произведением.

Входные данные

Первая строка входных данных содержит коэффициенты первого многочлена  $(a_0), (a_1), \dots, (a_n)$ .

Вторая строка содержит коэффициенты второго многочлена. Старшие коэффициенты обоих многочленов ненулевые.

Выходные данные

Выведите  $(m + n + 1)$  число — коэффициенты произведения данных многочленов от младшего к старшему

Пояснение к примеру

Пример соответствует выражению  $((-2 + x)(1 + 2x + x^2) = -2 - 3x + x^3)$ .

Примеры

Входные данные

-2 1

1 2 1

Выходные данные

-2 -3 0 1

---

# Слияние списков

## Задача S. Слияние списков

---

Даны два списка  $A$  и  $B$  упорядоченных по неубыванию. Объедините их в один упорядоченный список  $C$  (то есть он должен содержать  $\text{len}(A) + \text{len}(B)$  элементов). Решение оформите в виде функции  $\text{merge}(A, B)$ , возвращающей новый список. Алгоритм должен иметь сложность  $O(\text{len}(A) + \text{len}(B))$ . Модифицировать исходные списки запрещается. Использовать функцию `sorted` и метод `sort` запрещается.

Входные данные

Программа получает на вход два неубывающих списка, каждый в отдельной строке.

Выходные данные

Программа должна вывести последовательность неубывающих чисел, полученных объединением двух данных списков.

Примеры

Входные данные

1 5 7  
2 4 4 5

Выходные данные

1 2 4 4 5 5 7

---

# Пересечение множеств

## Задача Т. Пересечение множеств

---

Даны два списка, упорядоченных по возрастанию (каждый список состоит из различных элементов). Найдите пересечение множеств элементов этих списков, то есть те числа, которые являются элементами обоих списков. Алгоритм должен иметь сложность  $O(\text{len}(A) + \text{len}(B))$ .

Решение оформите в виде функции `Intersection(A, B)`. Функция должна возвращать список пересечения данных списков в порядке возрастания элементов. Модифицировать исходные списки запрещается.

Входные данные

Программа получает на вход два возрастающих списка, каждый в отдельной строке.

Выходные данные

Программа должна вывести последовательность возрастающих чисел, являющихся элементами обоих списков.

Примеры

Входные данные

1 3 4 7 9  
2 3 7 10 11

Выходные данные

3 7

---



# Решето Эратосфена

## Задача U. Решето Эратосфена

---

Определите  $N = 100000$  и создайте массив  $[\text{True}] * (N + 1)$ . Заполните его значениями так, чтобы  $\text{IsPrime}[i] == \text{True}$ , если  $i$  — простое число и  $\text{IsPrime}[i] == \text{False}$ , если  $i$  — составное.

Для этого сначала заполняем массив `True`. Затем “вычеркиваем” (то есть помечаем нулями) те элементы, которые делятся на 2, начиная с 4. Затем вычеркиваем те элементы, которые делятся на 3, начиная с 9. Затем вычеркиваем элементы, которые делятся на 5, начиная с 25... И так далее — находим следующее невычеркнутое число  $p$  и вычеркиваем все кратные  $p$  начиная с  $p^2$ .

В результате невычеркнутыми останутся только простые числа. Такая процедура называется “Решето Эратосфена”.

Напишите программу, которая строит решето Эратосфена, потом считывает натуральное число  $n$  и выводит  $n$ -е по счету простое число. Гарантируется, что это число не превосходит  $N$ .

Примеры

Входные данные

5

Выходные данные

11

---

# Шарики

## Задача V. Шарики

---

В одной компьютерной игре игрок выставляет в линию шарики разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарики при этом сдвигаются друг к другу, и ситуация может повториться. Если есть возможность удалить несколько групп, то удаляется самая левая из них.

Напишите программу, которая по данной ситуации определяет, сколько шариков будет сейчас уничтожено.

Входные данные

Программа получает на вход список чисел, задающих цвета шариков (числа от 0 до 9, каждому цвету соответствует свое целое число).

Выходные данные

Требуется вывести количество шариков, которое будет уничтожено.

Для удаления фрагмента списка удобно использовать срезы.

Примеры

Входные данные

1 3 3 3 2

Выходные данные

3

Входные данные

3 3 2 1 1 1 2 2 3 3

Выходные данные

10

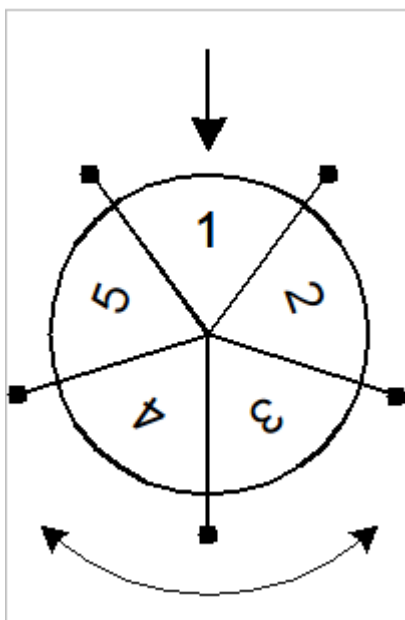
---

# Колесо Фортуны

## Задача W. Колесо Фортуны

Развлекательный телеканал транслирует шоу «Колесо Фортуны». В процессе игры участники шоу крутят большое колесо, разделенное на сектора. В каждом секторе этого колеса записано число. После того как колесо останавливается, специальная стрелка указывает на один из секторов. Число в этом секторе определяет выигрыш игрока.

Юный участник шоу заметил, что колесо в процессе вращения замедляется из-за того, что стрелка задевает за выступы на колесе, находящиеся между секторами. Если колесо вращается с угловой скоростью  $V$  градусов в секунду, и стрелка, переходя из сектора  $X$  к следующему сектору, задевает за очередной выступ, то текущая угловая скорость движения колеса уменьшается на  $K$  градусов в секунду. При этом если  $V \leq K$ , то колесо не может преодолеть препятствие и останавливается. Стрелка в этом случае будет указывать на сектор  $X$ .



Юный участник шоу собирается вращать колесо. Зная порядок секторов на колесе, он хочет заставить колесо вращаться с такой начальной скоростью, чтобы после остановки колеса стрелка указала на как можно большее число. Колесо можно вращать в любом направлении и придавать ему начальную угловую скорость от  $a$  до  $b$  градусов в секунду.

Требуется написать программу, которая по заданному расположению чисел в секторах, минимальной и максимальной начальной угловой скорости вращения колеса и величине замедления колеса при переходе через границу секторов вычисляет максимальный выигрыш.

Входные данные

Первая строка входного файла содержит целое число  $n$  — количество секторов колеса ( $3 \leq n \leq 100$ ).

Вторая строка входного файла содержит  $P$  положительных целых чисел, каждое из которых не превышает **1000** — числа, записанные в секторах колеса. Числа приведены в порядке следования секторов по часовой стрелке. Изначально стрелка указывает на первое число.

Третья строка содержит три целых числа:  $a, b$  и  $k$  ( $1 \leq a \leq b \leq 10^9, 1 \leq k \leq 10^9$ ).

#### Выходные данные

В выходном файле должно содержаться одно целое число — максимальный выигрыш.

#### Примечание

В первом примере возможны следующие варианты: можно придать начальную скорость колесу равную 3 или 4, что приведет к тому, что стрелка преодолеет одну границу между секторами, или придать начальную скорость равную 5, что позволит стрелке преодолеть 2 границы между секторами. В первом варианте, если закрутить колесо в одну сторону, то выигрыш получится равным 2, а если закрутить его в противоположную сторону, то — 5. Во втором варианте, если закрутить колесо в одну сторону, то выигрыш будет равным 3, а если в другую сторону, то — 4.

Во втором примере возможна только одна начальная скорость вращения колеса — 15 градусов в секунду. В этом случае при вращении колеса стрелка преодолеет семь границ между секторами. Тогда если его закрутить в одном направлении, то выигрыш составит 4, а если в противоположном направлении, то — 3.

Наконец, в третьем примере оптимальная начальная скорость вращения колеса равна 2 градусам в секунду. В этом случае стрелка вообще не сможет преодолеть границу между секторами, и выигрыш будет равен 5.

Правильные решения для тестов, в которых  $1 \leq a \leq b \leq 1000$ , будут оцениваться из 50 баллов.

#### Примеры

##### Входные данные

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

##### Выходные данные

```
5
```

##### Входные данные

```
5
1 2 3 4 5
15 15 2
```

##### Выходные данные

```
4
```

##### Входные данные

5

5 4 3 2 1

2 5 2

Выходные данные

5

---

# Наибольшее произведение двух чисел

## Задача X. Наибольшее произведение двух чисел

---

Дан список, заполненный произвольными целыми числами. Найдите в этом списке два числа, произведение которых максимально. Выведите эти числа в порядке неубывания.

Решение должно иметь сложность  $O(n)$ , где  $n$  - размер списка.

Примеры

Входные данные

4 3 5 2 5

Выходные данные

5 5

Входные данные

-4 3 -5 2 5

Выходные данные

-5 -4

---

# Наибольшее произведение трех чисел

## Задача Y. Наибольшее произведение трех чисел

---

В данном списке из  $n \leq 10^5$  целых чисел найдите три числа, произведение которых максимально.

Решение должно иметь сложность  $O(n)$ , где  $n$  - размер списка.

Выведите три искомых числа в любом порядке.

Примеры

Входные данные

3 5 1 7 9 0 9 -3 10

Выходные данные

10 9 9

Входные данные

-5 -30000 -12

Выходные данные

-5 -12 -30000

---

# Ханойская сортировка

## Задача Z. Ханойская сортировка

---

Кто-то перепутал Ханойскую головоломку и разместил диски на первом стержне не соблюдая правила игры. Переложите эти диски так, чтобы они оказались на одном из стержней строго в порядке возрастания номеров.

Входные данные

Программа получает на вход число дисков  $n \leq 10$ . Во второй строке записаны  $n$  чисел — номера дисков на первом стержне сверху вниз.

Выходные данные

Перемещать диск можно только в том случае, если он кладется на диск большего номера. Выведите последовательность перекладываний, размещающая диски на любом стержне в порядке возрастания номеров. Формат вывода одного перекладывания:  $ABC$ , где  $A$  номер перемещаемого диска ( $1 \leq A \leq n$ ),  $B$  — номер стержня с которого снимается диск,  $C$  — номер стержня на который кладется диск.

Примеры

Входные данные

3  
2 1 3

Выходные данные

2 1 2  
1 1 2  
3 1 3  
1 2 3  
2 2 1  
1 3 1  
3 3 2  
1 1 3  
2 1 2  
1 3 2