

Задание 25

https://yandex.ru/tutor/subject/problem/?problem_id=T4836

Дан массив, содержащий 2018 положительных целых чисел, не превышающих 15000. Необходимо уменьшить на удвоенное значение минимального элемента массива те элементы, которые более чем вдвое превышают этот минимум, и вывести изменённый массив по одному элементу в строке.

Например, для исходного массива из 5 элементов 40 100 27 90 54 программа должна вывести числа 40 46 27 36 54 по одному числу в строке (минимальный элемент исходного массива равен 27, поэтому элементы, *большие* 54, уменьшены на 54).

Напишите на одном из языков программирования программу для решения этой задачи.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

```
# кроме уже указанных
# допускается использование
# целочисленных переменных
# k, m
a = []
n = 2018
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа Вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, Free Pascal 2.6). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

Здесь метод **append** добавляет к списку **a** один элемент. Тело цикла повторяется **n** раз – от 0 до **n-1** (в Питоне правая граница не входит в указываемый диапазон). На каждой итерации цикла в изначально пустой список **a** (был инициализирован пустым списком **a = []**) последовательно добавляются элементы.

Решение

Данный тип задач можно отнести к достаточно простым, оценка за правильное решение этой задачи – 2 балла. Чтобы успешно решать такого рода задачи вы должны понимать, как использовать одномерный массив (список), логику операторов ветвления и цикла, арифметические операции с целыми числами. Залогом успеха в решении данной задачи является личный опыт написания такого рода программ на одном из алгоритмических языков программирования.

Приступаем к разбору.

Внимательно перечитайте условие задания и выделите основные моменты:

- дан **массив**,
- содержащий **2018 положительных целых чисел**,
- не превышающих **15000**,
- необходимо **уменьшить**
- на удвоенное значение **минимального элемента** массива
- те элементы, которые более чем вдвое **превышают этот минимум**,
- и вывести изменённый массив **по одному элементу в строке**.

Распишите приведённый текст задания в виде тезисов. Сделайте именно так, чтобы не упустить никакую деталь!

Итак, у нас есть последовательность из **n** элементов (массив, список) – значит будем запускать по нему цикл.

Элементы последовательности – это числа (целые и положительные) величиной не более **15000** – учтём это, может пригодится.

Некоторые из элементов последовательности нужно уменьшить.

На сколько уменьшить нам не дано – это нужно **сначала** вычислить. Сначала нужно узнать величину минимального элемента (это будет переменная **m**) и только потом уменьшить некоторые элементы на **2*m** (по условию задачи) – то есть из значения элемента **вычесть 2*m**.

После чего можно будет вывести массив по одному элементу в строке – то есть вывести все элементы в столбик – для этого тоже нужен цикл.

После анализа задания приходим к выводу, что можно организовать алгоритм решения, состоящий из трёх или из двух шагов.

Алгоритм, состоящий из трёх самостоятельных шагов:

- 1) ищем минимум;
- 2) уменьшаем некоторые значения, сохраняя в массиве на прежних позициях;
- 3) выводим элементы массива на экран в столбик.

Алгоритм, состоящий из двух самостоятельных шагов:

- 1) ищем минимум;
- 2) уменьшаем некоторые значения, сохраняя в массиве и одновременно выводя их на экран.

Во втором решении будет на один цикл меньше.

Итак, **шаг номер 1** – ищем минимум массива. Что для этого нужно сделать? Возможны два варианта: либо применить встроенный метод поиска минимума `min` (`m = min(a)` – это разрешается), либо написать самому цикл для перебора элементов массива (списка):

```
m = 15001
for i in range(0, n):
    if a[i] < m:
        m = a[i]
```

В данном случае цикл **for** перебирает все элементы списка **a** (индексы изменяются от 0 до n-1 – так как в Питоне правая граница не включается в диапазон). На каждом шаге проверяется текущий элемент **a[i]**, если он меньше чем **m** (а там мы храним текущий минимум), то минимум обновляем. Чтобы сравнение `if a[i] < m` работало и на первом шаге цикла в переменной **m** должно что-то быть – её нужно инициализировать каким-то значением. Обычно это бывает ноль, но только не в этом случае. Если до цикла в переменную **m** мы поместим 0, то условие `if a[i] < m` ни разу не будет истинным, так как по условию задачи все числа в массиве положительные – следовательно ни одно из них не может быть меньшим нуля. Чтобы поиск минимума увенчался успехом, нужно чтобы текущий элемент сравнивался с ранее выбранным минимумом, а у нас на первом шаге ещё не выбран минимум. Можно его назначить, например, записав туда любой из элементов массива (списка), например, первый (с индексом 0, так как индексация начинается с нуля) или просто записав число, заведомо большее, чем числа, указанные в условии задачи (можно записать 15001 или 15000 – оба варианта приведут к верному решению).

Итак, первая часть выполнена, теперь **шаг номер 2** – вывод ответов в столбик:

```
for i in range(0, n):
    if a[i] > 2*m:
        a[i] -= 2*m
    print(a[i])
```

- Обратите внимание, что тело второго цикла содержит два оператора:
- условный – он нужен, чтобы понять меняем ли мы текущий элемент массива или оставляем без изменений;
 - и команда вывода **print**, которая работает вне зависимости от условия и на каждой итерации цикла выводит значение текущего элемента в новой строке (в столбик).

Отмечу одну особенность, вот такая запись в Питоне:

```
a[i] -= 2*m
```

является сокращённой формой записи арифметической операции вычитания, если для вас это непривычно, то смело можете писать так:

```
a[i] = a[i] - 2*m
```

Это, собственно всё решение – мы добавили только 8 строк кода за два шага к исходной программе, далее привожу полный код:

```
a = []
n = 2018
for i in range(0, n):
    a.append(int(input()))

m = 15001
for i in range(0, n):
    if a[i] < m:
        m = a[i]

for i in range(0, n):
    if a[i] > 2*m:
        a[i] = a[i] - 2*m
    print(a[i])
```

Первая часть кода дана в условии задания, а заключительные 8 строчек кода мы дописали в качестве решения.