**Списки**

**Списки сохраняют данные**

**Пример 0: map нельзя перебрать в цикле больше 1 раза**

Дана последовательность чисел. Напечатать эту последовательность 2 раза.

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=1&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=1&up=0#sorted_table) |
| 3 7 -5 18 | 3 7 -5 18 3 7 -5 18 |

Как сделать? Проблема: **map** дает последовательность, которую можно перебрать в цикле только один раз.

m = map(int, input().split())

# напечатает 3 7 -5 18

for x in m:

print(x, end=' ') # печатать х, закончить печать пробелом

# ничего не напечатает,

# все числа из последовательности m уже прочитали

for x in m:

print(x, end=' ')

**Пример 1: list, печатаем последовательность 2 раза**

**Последовательность можно сохранить в памяти** и перебирать ее много раз. Элементы из последовательности не исчезают.

Для этого делаем из последовательности **list** (список).

m = map(int, input().split()) # получили последовательность

a = list(m) # сохранили последовательность в памяти

# печатаем первый раз

for x in a:

print(x, end=' ')

# печатаем второй раз

for x in a:

print(x, end=' ')

В питоне обычно пишут сразу сохранение последовательности в список:

a = list(map(int, input().split()))

Решим задачи.

**Задача 1 (Вьетнам): Сначала четные, потом нечетные**

Даны целые числа через пробел. Напечатать сначала четные числа, потом нечетные.

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=2&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=2&up=0#sorted_table) |
| 3 -6 7 -5 18 | -6 18 3 7 -5 |

**Задача 2 (Мьянма): Больше среднего**

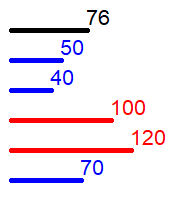
Даны целые числа на 1 строке через пробел. Найти среднюю длину.

Нарисовать среднюю длину черным и написать длину.

Нарисовать все отрезки. Если отрезок больше среднего, рисовать красным, иначе синим.

|  |
| --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=3&up=0#sorted_table) |
| 50 40 100 120 70 |

Среднее (50 + 40 + 100 + 120 + 70) / 5 = 380 / 5 = 76



**Создание списка и его печать**

Список можно создать с помощью функции **list** или перечислив элементы в [ ].

a = [3, 7, -5, 18] # создали список

print(a) # [3, 7, -5, 18]

print(\*a) # 3 7 -5 18

В пустом списке нет элементов. Его можно создать так:

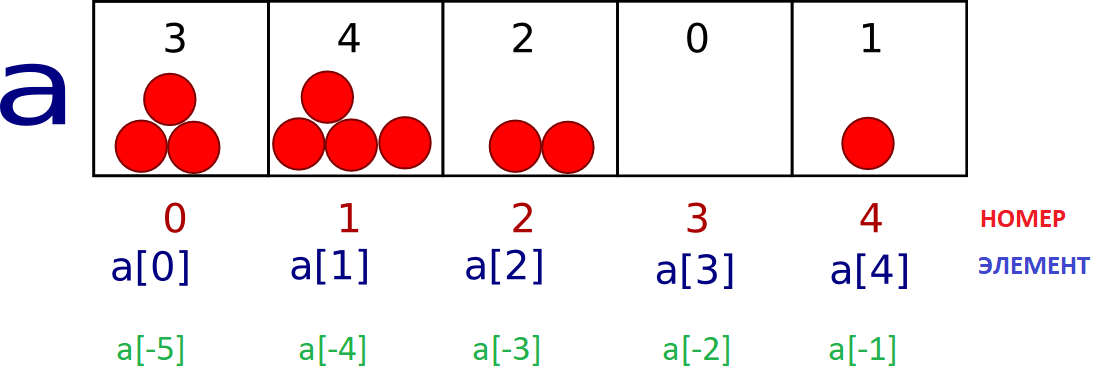
a1 = list() # пустой список, нет элементов

a2 = [] # тоже пустой список

**Элементы списка и его индексы**

Большая коробка поделена на 5 одинаковых **ячеек**. Коробка называется **a**. Все ячейки имеют **номер**: от 0 до 4. В ячейки положили шарики:

a = [3, 4, 2, 0 1]

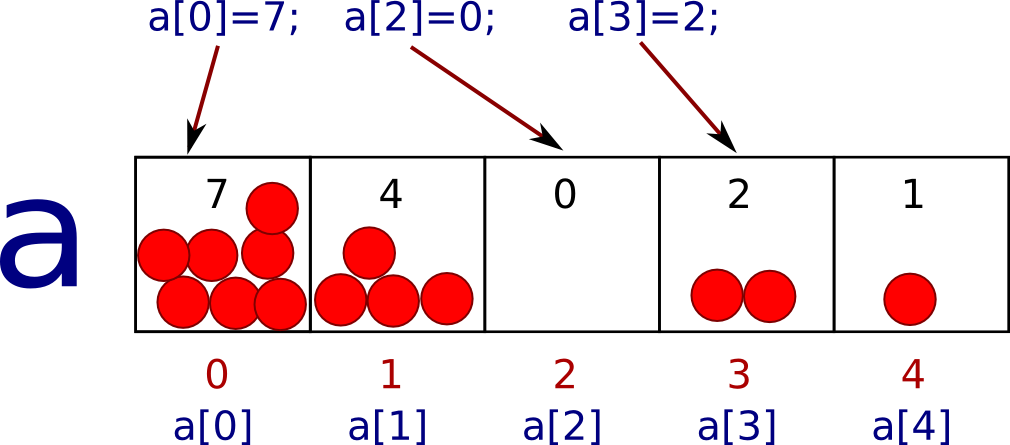


В каждую ячейку можно положить **число** мячей.

a[0] = 7

a[2] = 0

a[3] = 2



Какое число в ячейке номер 3?

a[3] = a[0] + 2

**Отрицательные номера** - идем от конца в начало списка.

a[-1] = 8 # в ПОСЛЕДНЮЮ ячейку положили число 8

**Длина списка**

Функция **len** считает, сколько элементов в списке.

a = [3, 7, -5, 18] # создали список

print(a) # [3, 7, -5, 18]

n = len(a) # в n записали длину списка a

print(n) # 4

**Перебираем элементы списка**

Для перебора всех элементов списка используют цикл **for**.

Переберем список a = [3, 7, -5, 18]

**Перебор по элементам**

Если нужны только элементы, а номера не нужны, то перебираем так:

for x in a:

print(x)

Напечатает:

3

7

-5

18

**Перебор только по номерам**

Если нужен номер, а элемент мы получим как a[i], то нужен **range**

for i in range(len(a)):

print(i, a[i])

Напечатает:

0 3

1 7

2 -5

3 18

**Перебор по элементам и номерам**

Если нужны и элемент, и его номер, то нужен **enumerate** :

for i, x in enumerate(a):

print(i, x)

Напечатает:

0 3

1 7

2 -5

3 18

**range**

|  |  |  |
| --- | --- | --- |
| [**range**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=4&up=0#sorted_table) | [**получим**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=4&up=0#sorted_table) | [**что это**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=2&table=4&up=0#sorted_table) |
| range(5) | 0 1 2 3 4 | целые числа от 0 до 5, НЕ включая 5 |
| range(2, 5) | 2 3 4 | целые числа от 2 до 5, включая 2, НЕ включая 5 |
| range(1, 11, 2) | 1 3 5 7 9 | целые числа от 1 до 11 с шагом 2, включая 1, НЕ включая 11 |

**Задачи (Мьянма)**

На прямой 0Х есть точки. Их x координаты заданы в 1 строку через пробел.

**Задача 3. Нарисовать все точки**

Нарисовать все точки синим цветом. Написать около каждой точки ее номер.

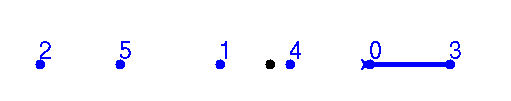
|  |
| --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=5&up=0#sorted_table) |
| 100 -50 -230 180 20 -150 |



**Задача 4. Расстояние от точки с номером 0 до точки с номером i**

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=6&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=6&up=0#sorted_table) |
| 100 -50 -230 180 20 -150  3 | 80 |

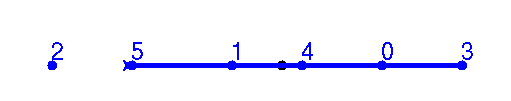
* Дано
  + Даны точки на 1 строке.
  + На следующей строке дано i - НОМЕР точки.
* Напечатать
  + Расстояние от точки с номером 0 до точки с номером i.
* Нарисовать
  + Все точки синим цветом. Написать около каждой точки ее номер.
  + Отрезок от точки с номером 0 до точки с номером i



**Задача 5. Расстояние от точки с номером k до точки с номером i**

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=7&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=7&up=0#sorted_table) |
| 100 -50 -230 180 20 -150  3 5 | 230 |

* Дано
  + Даны точки на 1 строке.
  + На следующей строке дано k - НОМЕР точки.
  + На следующей строке дано i - НОМЕР точки.
* Напечатать
  + Расстояние от точки с номером k до точки с номером i.
* Нарисовать
  + Все точки синим цветом. Написать около каждой точки ее номер.
  + Отрезок от точки с номером k до точки с номером i



**Задача 6. Самая правая точка (max)**

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=8&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=8&up=0#sorted_table) |
| 100 -50 -230 180 20 -150 | 180 3 |

* Дано
  + Даны точки на 1 строке.
* Напечатать
  + Координату самой правой точки и ее номер
* Нарисовать
  + Все точки синим цветом.
  + Самую правую точку красным цветом.
  + Написать около каждой точки ее номер.

**Задача 7. Самая левая точка (min)**

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=9&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=9&up=0#sorted_table) |
| 100 -50 -230 180 20 -150 | -230 2 |

* Дано
  + Даны точки на 1 строке.
* Напечатать
  + Координату самой правой точки и ее номер
* Нарисовать
  + Все точки синим цветом.
  + Самую правую точку красным цветом.
  + Написать около каждой точки ее номер.

**EFACTOR Задача про купца**

**Задача: поменять в списке местами 2 числа**

Даны числа через пробел и номера i и j. Поменять местами в списке числа с номерами i и j.

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=10&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=10&up=0#sorted_table) |
| 3 2 14 0 2 | [14, 2, 3] |
| 3 2 14 5 22  2 3 | [3, 2, 5, 14, 22] |
| 3 2 14 5 22  0 -1 | [22, 2, 14, 5, 3] |

**Задача: делим конфеты**

У людей есть конфеты.

Люди имеют номера 0, 1, 2, 3 и дальше.

Даны числа - сколько конфет у людей.

Даны два номера - эти два человека хотят сделать их конфеты равными.

Если равные не получаются, то лишнюю конфету они отдают.

Написать функцию delim(i, j), которая делит конфеты поровну для человека i и человека j.

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=11&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=11&up=0#sorted_table) |
| 3 5 17 0 1 | [4, 4, 17] |
| 3 2 14 5 22  0 1 | [2, 2, 14, 5, 22] |
| 3 2 14 5 22  0 -1 | [12, 2, 14, 5, 12] |

**Задача: проверить, что все числа одинаковые**

Даны числа через пробел. Напечатать YES, если все числа одинаковые. Иначе напечатать NO.

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=12&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=12&up=0#sorted_table) |
| 2 2 2 2 2 | YES |
| 1 1 4 | NO |

**Задача: сделать равным по парам**

У людей есть конфеты.

Люди имеют номера 0, 1, 2, 3 и дальше.

Люди парами умеют делить конфеты с помощью функции delim(i, j).

Поделите конфеты по парам: человек 0 и 1, человек 1 и 2, человек 2 и 3 и дальше.

Последним поделите конфеты первого человека a[0] и последнего человека a[-1].

После каждой пары печатайте номера людей и все конфеты.

|  |
| --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=13&up=0#sorted_table) |
| 3 2 1 3 5 |

[3, 2, 1, 3, 5]

0 1 [2, 2, 1, 3, 5]

1 2 [2, 1, 1, 3, 5]

2 3 [2, 1, 2, 2, 5]

3 4 [2, 1, 2, 3, 3]

0 -1 [2, 1, 2, 3, 2]

**Задача: купец**

Купец нанял N работников. Все работники делали одинаковую работу. По окончании работ купец расплатился с работниками монетами. Но он дал каждому монет не поровну, а как-нибудь: кому-то много, а кому-то мог вообще не заплатить.

Работники возмутились и стали требовать справедливости.

Купец обещал уладить дело.

Он поставил всех работников в круг. Затем у каждой пары работников он брал деньги и складывал. Если число монет двух работников нечетное, то одну монету купец забирал себе, а остальное поровну делил между этими двумя работниками. Купец начинал круг с пары 1-2, затем 2-3, ... и заканчивал круг парой N-1

После каждого такого "уравнивания" все проверяли поровну ли у них монет. Если у всех было поровну, процесс заканчивался.

Рабочие, получив РАВНОЕ количество монет уходили домой, а купец получал еще немножечко денег.

Написать программу, которая вычисляет сколько монет получил купец и сколько каждый работник.

Дано: число монет, которое вначале получил каждый работник.

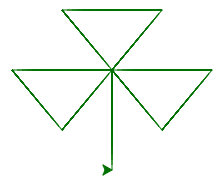
Найти: Два целых числа: количество монет, полученных купцом и количество монет, которое осталось у каждого работника.

|  |  |
| --- | --- |
| [**input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=14&up=0#sorted_table) | [**output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=14&up=0#sorted_table) |
| 2 2 2 2 | 0 2 |
| 3 3 1 3 3 | 3 2 |

**Пример 2. Ломанная одним цветом**

Даны x, y координаты точки. По 1 точке на строку. 11 точек. Нарисовать линию от первой точки до последней.

|  |
| --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=15&up=0#sorted_table) |
| 0 0 50 -60 100 0 0 0 50 60 -50 60 0 0 -100 0 -50 -60 0 0 0 -100 |



Решение (list не нужен, хватит последовательности):

import turtle

import sys

fp = sys.stdin

t = turtle.Turtle()

t.color('darkgreen')

for string in fp: # для всех строк

x, y = map(int, string.split()) # прочитали x и y для 1 точки

t.setpos((x, y)) # передвинули черепаху в точку (x, y)

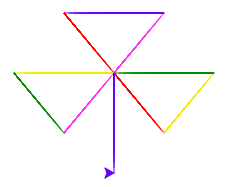
turtle.done()

**Пример 3. Ломанная разными цветами**

Даны x, y координаты точки. По 1 точке на строку. Всего 11 точек.

Нарисовать линию от первой точки до последней, менять цвет отрезка: "blue", "red", "gold", "green", "violet".

|  |
| --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=16&up=0#sorted_table) |
| 0 0 50 -60 100 0 0 0 50 60 -50 60 0 0 -100 0 -50 -60 0 0 0 -100 |



Создадим список цветов colors и будем брать следующий цвет colors[i]

import turtle

import sys

fp = sys.stdin

t = turtle.Turtle()

t.color('darkgreen')

colors = [

"blue", # colors[0]

"red", # colors[1]

"gold", # colors[2]

"green", # colors[3]

"violet" # colors[4]

]

i = 0 # НОМЕР первого цвета 0

for string in fp: # для всех строк

t.color(colors[i]) # берем цвет по его номеру i

x, y = map(int, string.split()) # прочитали x и y для 1 точки

t.setpos((x, y)) # передвинули черепаху в точку (x, y)

turtle.done()

Программа не работает. Точек больше, чем цветов. Когда пытаемся взять colors[5], его нет. Что делать?

Надо брать цвета с номерами 0, 1, 2, 3, 4 и опять 0, 1, 2, 3, 4 и опять 0...

Придумаем формулу и проверим ее:

|  |  |
| --- | --- |
| [**i**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=17&up=0#sorted_table) | [**i%5**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=1&table=17&up=0#sorted_table) |
|  |  |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | `0 |
| 6 | 1 |
| 7 | 2 |

5 - это длина списка colors, длину тоже лучше посчитать в программе, а не писать число 5.

import turtle

import sys

fp = sys.stdin

t = turtle.Turtle()

t.color('darkgreen')

colors = [

"blue", # colors[0]

"red", # colors[1]

"gold", # colors[2]

"green", # colors[3]

"violet" # colors[4]

]

n = len(colors) # длина списка 5, сохранили в n

i = 0 # НОМЕР первого цвета 0

for string in fp: # для всех строк

t.color(colors[i%n]) # берем цвет по его номеру i

p = map(int, string.split()) # прочитали p = (x,y) для 1 точки

t.setpos(p) # передвинули черепаху в точку (x, y)

turtle.done()

**Читаем числа и сохраняем их в список**

**Целые числа в одну строку**

|  |
| --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=18&up=0#sorted_table) |
| 3 7 -5 18 |

a = list(map(int, input().split()))

print(a)

**Целые числа в много строк, 1 число на 1 строке**

|  |
| --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=19&up=0#sorted_table) |
| 3 7 -5 18 |

import sys

fp = sys.stdin

a = [int(string) for string in fp] # берем по 1 строке, из каждой строки делаем int

print(a)

**x, y координата на 1 строке, много строк**

|  |
| --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList1?sortcol=0&table=20&up=0#sorted_table) |
| 30 -50 100 150 200 0 |

import sys

fp = sys.stdin

a = [tuple(map(int, string.split())) for string in fp] # берем по 1 строке, из каждой строки делаем (x, y)

print(a) # [(30, -50), (100, 150), (200, 0)]

print(a[0]) # (30, -50)

x, y = a[0]

print(x) # 30

print(y) # 50

**Задачи (Вьетнам)**

**Задача 3a. Далекая точка**

Даны точки.

Нарисовать точку (0, 0) черным.

Нарисовать точки синим. Самую далекую от (0,0) точку красным.

30 -50

100 150

200 0

**Задача 3b. Далекая точка - много точек**

Даны точки.

Нарисовать точку (0, 0) черным.

Нарисовать точки синим. Самую далекую от (0,0) точку красным. Если таких точек много, все их нарисовать красным.

**Задача 3с. Далекая точка - много точек соединить отрезком**

Даны точки.

Нарисовать точку (0, 0) черным.

Нарисовать точки синим. Самую далекую от (0,0) точку красным. Если таких точек много, все их нарисовать красным и соединить отрезками.

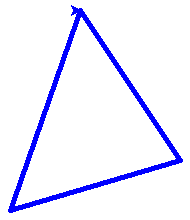
**Задача 4. Многоугольник**

Даны точки. Нарисовать по ним многоугольник.

30 -50

100 150

200 0



**Задача 5. Где ближайшая точка**

2

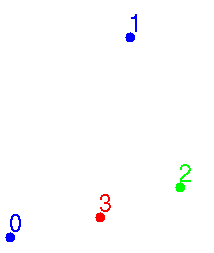
30 -50

100 150

200 0

120 -30

* Дано:
  + номер точки k
  + x y координаты точек по 1 точке на строку
* Напечатать:
  + расстояние до ближайшей точке к точке с номером k (k-тая точка)
  + номер ближайшей точки, ближайшей к k-той точке.
* Нарисовать:
  + k-тую точку зеленым
  + ближайшую к ней - красным
  + остальные точки - синим
  + у каждой точки написать ее номер

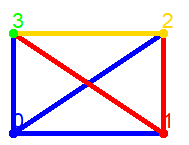


**Задача 5. Соединить все точки**

Даны точки. Нарисовать точки и написать их номера.

Соединить точки всеми возможными линиями.

Для этого написать функцию **connect(a, i)**, которая для точки a[i] рисует все соединения, которых еще нет.



**Задача 6. Две ближайшие точки**

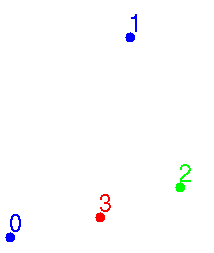
30 -50

100 150

200 0

120 -30

* Дано:
  + x y координаты точек по 1 точке на строку
* Напечатать:
  + расстояние для каждых двух точек
  + номера точек между которыми самое маленькое расстояние
* Нарисовать:
  + точки - синим
  + у каждой точки написать ее номер
  + две точки с самым меньшим расстоянием - красным. Соединить их



**Задача 6. Шары - каких больше**

10 ячеек заполнены красными и зелеными шарами. Вес красных шаров обозначается положительными целым числом, вес зеленых отрицательным.

Написать программу, которая определяет какие шары тяжелее: все красные или все зеленые.

В ответе писать: RED, GREEN или ==

**Задача 7. Шары - отделить красные от зеленых**

10 ячеек заполнены красными и зелеными шарами. Вес красных шаров обозначается положительными целым числом, вес зеленых отрицательным.

Написать функцию **greenred(a)** , которая перемещает в списке **a** все зеленые шары налево, а красные направо.

-- [TatyanaDerbysheva](http://acm.mipt.ru/twiki/bin/view/Main/TatyanaDerbysheva) - 12 Oct 2018