

**Сдать задание нужно до 17 октября 2015г. включительно.**

**Задача № 1 (2 балла)**

**1\_1.** Найти, на сколько нулей оканчивается  $n! = 1 * 2 * 3 * \dots * n$ .

$n \leq 1000$ .

in	out
25	6

**1\_2.** Вывести разложение натурального числа  $n$  на простые множители. Простые множители должны быть упорядочены по возрастанию и разделены пробелами.

$2 \leq n \leq 10^6$ .

in	out
75	3 5 5

**1\_3.** Даны две рациональные дроби:  $a/b$  и  $c/d$ . Сложить их и результат представьте в виде несократимой дроби  $m/n$ . Вывести числа  $m$  и  $n$ .

$a, b, c, d \leq 1000$ .

in	out
3 10 5 18	26 45

**1\_4.** Дано натуральное число  $N$ . Представить  $N$  в виде  $A + B$ , так, что НОД( $A, B$ ) максимален,  $A \leq B$ . Вывести  $A$  и  $B$ . Если возможно несколько ответов - вывести ответ с минимальным  $A$ .

$n \leq 10^7$ .

in	out
35	7 28

**1\_5.** Вывести квадраты натуральных чисел от 1 до  $n$ , используя только  $O(n)$  операций сложения и вычитания (умножением пользоваться нельзя).

$n \leq 1000$ .

in	out
5	1 4 9 16 25

**1\_6.** Дан массив целых чисел  $A[0..n]$ . Не используя других массивов переставить элементы массива  $A$  в обратном порядке за  $O(n)$ .

$n \leq 10000$ .

in	out
4 3 9 -5 2	2 -5 9 3

**1\_\***. Найти все простые числа в диапазоне  $[2..n]$  за  $O(n)$ .

in	out
15	2 3 5 7 11 13

**Задача № 2 (3 балла)**

В каждой задаче, где начальными данными является массив вначале вводится количество элементов, а затем и сами элементы массива.

**2\_1.** Даны два массива целых чисел одинаковой длины  $A[0..n-1]$  и  $B[0..n-1]$ . Необходимо найти первую пару индексов  $i_0$  и  $j_0$ ,  $i_0 \leq j_0$ , такую что  $A[i_0] + B[j_0] = \max \{A[i] + B[j], \text{ где } 0 \leq i < n, 0 \leq j < n, i \leq j\}$ . Время работы -  $O(n)$ .

$n \leq 100000$ .

in	out
4 4 -8 6 0 -10 3 1 1	0 1

**2\_2.** Вычислить площадь выпуклого n-угольника, заданного координатами своих вершин. Вначале вводится количество вершин, затем последовательно целочисленные координаты всех вершин в порядке обхода против часовой стрелки.

$n < 1000$ , координаты  $< 10000$ .

Указание. Для вычисления площади n-угольника можно посчитать сумму ориентированных площадей трапеций под каждой стороной многоугольника.

in	out
3 0 1 1 0 2 2	1.5

**2\_3.** Даны два строго возрастающих массива целых чисел  $A[0..n]$  и  $B[0..m]$  и число  $k$ . Найти количество таких пар индексов  $(i, j)$ , что  $A[i] + B[j] = k$ . Время работы  $O(n + m)$ .

$n, m \leq 100000$ .

Указание. Обходите массив  $B$  от конца к началу.

in	out
4 -5 0 3 18 5 -10 -2 4 7 12 7	3

**2\_4.** “Считалочка”. В круг выстроено  $N$  человек, пронумерованных числами от 1 до  $N$ . Будем исключать каждого  $k$ -ого до тех пор, пока не уцелеет только один человек. (Например, если  $N=10$ ,  $k=3$ , то сначала умрет 3-й, потом 6-й, затем 9-й, затем 2-й, затем 7-й, потом 1-й, потом 8-й, за ним - 5-й, и потом 10-й. Таким образом, уцелеет 4-й.) Необходимо определить номер уцелевшего.

$N, k \leq 10000$ .

in	out
10 3	4

**2\_\*** Дан массив целых чисел  $A[0..n]$ . Массив произвольным образом заполнен натуральными числами из диапазона  $[0..n - 1]$ . Одно или несколько значений в массиве может повторяться. Необходимо найти любой повтор за  $O(n)$ , памяти  $O(1)$ . Исходный массив хранить можно, модифицировать нельзя.

$n \leq 10000$ .

in	out
8 1 2 4 5 6 1 0 3	1

### Задача № 3 (4 балла)

**3\_1.** Дан отсортированный массив целых чисел  $A[0..n-1]$  и массив целых чисел  $B[0..m-1]$ . Для каждого элемента массива  $B[i]$  найдите минимальный индекс  $k$  минимального элемента массива  $A$ , равного или превосходящего  $B[i]$ :  $A[k] \geq B[i]$ . Если такого элемента нет, выведите  $n$ . Время работы поиска  $k$  для каждого элемента  $B[i]$ :  $O(\log(k))$ .

$n, m \leq 10000$ .

Формат входных данных.

В первой строчке записаны числа  $n$  и  $m$ . Во второй и третьей массивы  $A$  и  $B$  соответственно.

in	out
2 1 1 2 2	1
4 3 2 4 5 7 4 6 1	1 3 0

**3\_2.** Дан массив целых чисел  $A[0..n-1]$ . Известно, что на интервале  $[0, m]$  значения массива строго возрастают, а на интервале  $[m, n-1]$  строго убывают. Найти  $m$  за  $O(\log m)$ .

$n \leq 10000$ .

in	out
10 1 2 3 4 5 6 7 6 5 4	6

**3\_3.** Даны два массива неповторяющихся целых чисел, упорядоченные по неубыванию.  $A[0..n-1]$  и  $B[0..m-1]$ .  $n \gg m$ . Найдите их пересечение. Требуемое время работы:  $O(m * \log k)$ , где  $k$  - позиция элемента  $B[m-1]$  в массиве  $A$ . В процессе поиска очередного элемента  $B[i]$  в массиве  $A$  пользуйтесь результатом поиска элемента  $B[i-1]$ .

$n, k \leq 10000$ .

in	out
5 3 1 2 3 4 5 1 3 5	1 3 5

**3\_4.** Дан отсортированный массив целых чисел  $A[0..n-1]$  и массив целых чисел  $B[0..m-1]$ . Для каждого элемента массива  $B[i]$  найдите минимальный индекс элемента массива  $A[k]$ , ближайшего по значению к  $B[i]$ . Время работы поиска для каждого элемента  $B[i]$ :  $O(\log(k))$ .

$n \leq 100000, m \leq 1000$ .

in	out
3 10 20 30 3 9 18 35	0 1 2
3 10 20 30 4 8 9 10 32	0 0 0 2

### Задача № 4 (4 балла)

Во всех задачах из следующего списка следует написать структуру данных, обрабатывающую команды `push*` и `pop*`.

Формат входных данных.

В первой строке количество команд  $n$ .  $n \leq 100000$ .

Каждая команда задаётся как 2 целых числа:  $a$   $b$ .

$a = 1$  - push front

$a = 2$  - pop front

$a = 3$  - push back

$a = 4$  - pop back

Для очереди используются команды 2 и 3. Для дека используются все четыре команды.

Если дана команда pop\*, то число  $b$  - ожидаемое значение. Если команда pop вызвана для пустой структуры данных, то ожидается "-1".

Формат выходных данных.

Требуется напечатать YES - если все ожидаемые значения совпали. Иначе, если хотя бы одно ожидание не оправдалось, то напечатать NO.

**4\_1.** Реализовать очередь с динамическим зацикленным буфером.

in	out
3 3 44 3 50 2 44	YES
2 2 -1 3 10	YES
2 3 44 2 66	NO

**4\_2.** Реализовать дек с динамическим зацикленным буфером.

in	out
3 1 44 3 50 2 44	YES
2 2 -1 1 10	YES
2 3 44 4 66	NO

**4\_3.** Реализовать очередь с помощью двух стеков. Использовать стек, реализованный с помощью динамического буфера.

in	out
3 3 44	YES

3 50 2 44	
2 2 -1 3 10	YES
2 3 44 2 66	NO

**4\_\***. Реализовать очередь при помощи нескольких стеков. Каждая операция `pop front` и `push back` должна выполняться за  $O(1)$ .

### Задача № 5 (4 балла)

Решение всех задач данного раздела предполагает использование стека. Способ реализации стека может быть любым (список/динамический массив).

#### 5\_1. Скобочная последовательность.

Дан фрагмент последовательности скобок, состоящей из символов `(){}[]`.

Требуется определить, возможно ли продолжить фрагмент в обе стороны, получив корректную последовательность.

Длина исходной последовательности  $\leq 200000$ .

Формат входных данных. Строка, содержащая символы `(){}[]` и, возможно, перевод строки.

Формат выходных данных. Если возможно - вывести минимальную корректную последовательность, иначе - напечатать **"IMPOSSIBLE"**.

in	out
{[[[[{[]	{[[[[{[]]]]
{[[[[{[]	IMPOSSIBLE
]{}[](([]	{[]{}[](([])

#### 5\_2. Стековые анаграммы.

Пара слов называется стековой анаграммой, если одно слово можно получить из другого, проведя последовательность стековых операций с его буквами (взять очередную букву исходного слова и поместить ее в стек; взять букву из стека и добавить ее в конец выходного слова).

Для заданной пары слов требуется определить, можно ли выполнить последовательность стековых операций, переводящую первое слово во второе. Все буквы в слове различные.

Длина анаграммы  $\leq 10000$ .

Формат входных данных. Пара слов, являющихся анаграммой.

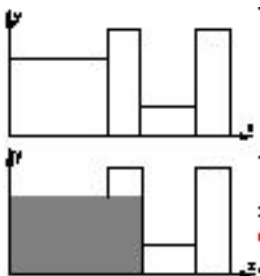
Формат выходных данных. **YES**, если последовательность стековых операций существует и **NO** в противном случае.

in	out
STOL SLOT	YES
ABC CAB	NO

#### 5\_3. Прямоугольники.

Дана последовательность  $N$  прямоугольников различной ширины и высоты  $(w_i, h_i)$ . Прямоугольники

расположены, начиная с точки (0, 0), на оси OX вплотную друг за другом (вправо). Требуется найти  $M$  - площадь максимального прямоугольника (параллельного осям координат), который можно вырезать из этой фигуры.



Время работы -  $O(n)$ .

Формат входных данных. В первой строке задано число  $N$  ( $1 \leq N \leq 10000$ ). Далее идет  $N$  строк. В каждой строке содержится два числа width и height: ширина и высота  $i$ -го прямоугольника. ( $0 \leq \text{width} \leq 10000$ ,  $0 \leq \text{height} \leq 10000$ )

Формат выходных данных. вывести число  $M$ . ( $0 \leq M \leq 10^9$ ).

in	out
4 30 30 10 40 20 10 10 40	1200
1 1 3000	3000
5 10 10 10 20 10 30 10 40 10 10	600

#### 5\_4. Вычисление выражения.

Дано выражение в инфиксной записи. Вычислить его, используя перевод выражения в постфиксную запись.

Выражение не содержит отрицательных чисел.

Количество операций  $\leq 100$ .

Формат входных данных. Строка, состоящая из символов "0123456789-+\*/()"

Гарантируется, что входное выражение корректно, нет деления на 0, вычислимо в целых числах. Деление целочисленное.

Формат выходных данных.

Значение выражения.

in	out
1 + 2	3
200-(123+34*2)+(48-2)	55

### Задача № 6 (4 балла)

Дано  $N$  кубиков. Требуется определить каким количеством способов можно выстроить из этих кубиков

пирамиду.

$N \leq 200$ .

Формат входных данных:

На вход подается количество кубиков  $N$ .

Формат выходных данных:

Вывести число различных пирамид из  $N$  кубиков.

**6\_1. Высокая пирамида.** Каждый вышележащий слой пирамиды должен быть не больше нижележащего.

in	out
3	3
5	7
7	15

**6\_2. Широкая пирамида.** Каждый вышележащий слой пирамиды должен быть строго меньше нижележащего.

in	out
3	2
5	3
7	5

## Задача № 7 (3 балла)

**7\_1. Атлеты.**

В город  $N$  приехал цирк с командой атлетов. Они хотят удивить горожан города  $N$  — выстроить из своих тел башню максимальной высоты. Башня — это цепочка атлетов, первый стоит на земле, второй стоит у него на плечах, третий стоит на плечах у второго и т.д.

Каждый атлет характеризуется силой  $s_i$  (kg) и массой  $m_i$  (kg). Сила — это максимальная масса, которую атлет способен держать у себя на плечах.

К сожалению ни один из атлетов не умеет программировать, так как всю жизнь они занимались физической подготовкой, и у них не было времени на изучение языков программирования. Помогите им, напишите программу, которая определит максимальную высоту башни, которую они могут составить.

Известно, что если атлет тяжелее, то он и сильнее:

если  $m_i > m_j$ , то  $s_i > s_j$ .

Атлеты равной массы могут иметь различную силу.

Формат входных данных:

Вход содержит только пары целых чисел — массу и силу атлетов. Число атлетов  $1 \leq n \leq 100000$ . Масса и сила являются положительными целыми числами меньше, чем 2000000.

Формат выходных данных:

Выход должен содержать натуральное число — максимальную высоту башни.

in	out
3 4 2 2 7 6 4 5	3

**7\_2. Заявки на переговоры.**

В большой IT-фирме есть только одна переговорная комната. Желающие посоветоваться заполняют заявки с желаемым временем начала и конца. Ваша задача определить максимальное количество заявок, которое может быть удовлетворено.

Число заявок  $\leq 100000$ .

Формат входных данных:

Вход содержит только пары целых чисел — начала и концы заявок.

Формат выходных данных:

Выход должен содержать натуральное число — максимальное число заявок.

in	out
1 2 2 5 7 10 2 3 6 7 4 7 3 6 3 5 1 2 4 5	5