# Приложение №1

#

# Состав заданий

Конкурсные задания состоят из трех частей:

* Задания на программирование:
	+ Задача №1 – 5 баллов,
	+ Задача №2 – 5 баллов,
	+ Задача №3 – 5 баллов,
	+ Задача №4 – 5 баллов,
	+ Задача №5 – 10 баллов,
	+ Задача №6\* – 10 баллов,
	+ Задача №7\* – 10 баллов,
	+ Задача №8\* – 10 баллов,
	+ Задача №9\* – 15 баллов,
	+ Задача №10\* – 25 баллов,
* Сопроводительное письмо – 10 баллов;
* Тест по направлению «Спутниковая навигация» – 25 баллов;

## Задание №1 - 5 баллов

Напишите программу, которая найдёт три ключа с самыми высокими значениями в словаре my\_dict = {'a':500, 'b':5874, 'c': 560,'d':400, 'e':5874, 'f': 20}.

## Задание №2 – 5 баллов

Напишите проверку на то, является ли строка палиндромом. Палиндром — это слово или фраза, которые одинаково читаются слева направо и справа налево.

## Задание №3 – 5 баллов

Напишите программу, которая находит точки пересечения двух окружностей. Программа на вход принимает две строки, которые включают в себя координаты центров окружностей и их радиусы, выводит координаты точек пересечения, либо при отсутствии пересечения “None”.

**Пример ввода:**

x1 y1 r1

x2 y2 r2

## Задание №4 – 5 баллов

Напишите программу, которая рассчитывает количество яиц, приносимых курятником. Каждая курица приносит каждые 2 дня по яйцу. На вход программа принимает количество куриц, петухов и прошедших дней. Выводит на экран количество яиц.

Пример ввода: строка формата “N M K”, где N - количество куриц, M - количество петухов, K-количество дней

## Задание №5 – 10 баллов

Напишите программу для решения классической древнекитайской головоломки:

Мы насчитали N голов и M ног у кур и кроликов на ферме. Сколько у нас кроликов и сколько цыплят?

Программа на вход должна принимать строку в формате: “N M”

## Задание №6\* – 10 баллов

На вход подается массив **arr** целых чисел, напишите программу, которая проверяет существование двух индексов **i** и **j**, такие, что :

**i != j**

**0 <= i, j < arr.length**

**arr[i] == 2 \* arr[j]**

**Пример 1:**

**Input**: arr = [10,2,5,3]

**Output**: true Explanation: For i = 0 and j = 2, arr[i] == 10 == 2 \* 5 == 2 \* arr[j]

**Пример 2:**

**Input**: arr = [3,1,7,11]

**Output**: false Explanation: There is no i and j that satisfy the conditions.

**Ограничения**:

2 <= arr.length <= 500

-103 <= arr[i] <= 103

## Задание №7\* – 10 баллов

На вход подается массив **arr** целых чисел фиксированного размера. Напишите программу, которая удваивает каждый ноль, сдвигая оставшиеся элементы вправо.

Важно: элементы, выходящие за длину массива, не записываются.

**Пример 1**:

**Input**: arr = [1,0,2,3,0,4,5,0]

**Output**: [1,0,0,2,3,0,0,4]

**Пример 2**:

**Input**: arr = [1,2,3]

**Output**: [1,2,3]

**Ограничения**:

1 <= arr.length <= 104

0 <= arr[i] <= 9

## Задание №8\* – 10 баллов

Напишите программу, которая принимает на вход строку чисел, удаляет повторяющиеся числа и выводит строку без повторяющихся чисел.

Пример ввода: 12,24,35,24,88,120,155,88,120,155

## Задание №9\* – 15 баллов

Написать программу, которая сортирует элементы матрицы. Программа принимает на вход число строк матрицы и строки матрицы, на выход выдаёт отсортированную матрицу.

Пример ввода:

3

3,5,2

1,2,7

9,8,1

Пример вывода:

1,1,2

2,3,5

7,8,9

## Задание №10\* – 25 баллов

В самолете n рядов и по три кресла слева и справа в каждом ряду. Крайние кресла (A и F) находятся у окна, центральные (C и D) – у прохода. На регистрацию приходят группы из одного, двух или трех пассажиров. Они желают сидеть рядом, то есть на одном ряду и на одной стороне: левой или правой. Например, группа из двух пассажиров может сесть на кресла B и C, но не может сесть на кресла C и D, потому что они разделены проходом, а также не может сесть на кресла A и C, потому что тогда они окажутся не рядом. Кроме того, один из пассажиров каждой группы очень требовательный – он хочет сесть либо у окна, либо у прохода. Конечно же, каждая группа из пассажиров хочет занять места в ряду с как можно меньшим номером, ведь тогда они скорее выйдут из самолета после посадки. Для каждой группы пассажиров определите, есть ли места в самолете, подходящие для них.

**Формат ввода**

Первая строка содержит число n (1 ≤ n ≤ 100) – количество рядов в самолете. Далее в n строках вводится изначальная рассадка в самолете по рядам (от первого до n-го), где символами. (точка) обозначены свободные места, символами # (решетка) обозначены занятые места, а символами \_ (нижнее подчеркивание) обозначен проход между креслами C и D каждого ряда. Следующая строка содержит число m (1 ≤ m ≤ 100) – количество групп пассажиров. Далее в m строках содержатся описания групп пассажиров. Формат описания такой: num side position, где num – количество пассажиров (число 1, 2 или 3), side – желаемая сторона самолета (строка left или right), position – желаемое место требовательного пассажира (строка aisle или window).

**Формат вывода**

Если группа может сесть на места, удовлетворяющие ее требованиям, то выведите строку Passengers can take seats: и список их мест в формате row letter, упорядоченный по возрастанию буквы места. Затем выведите в n строках получившуюся рассадку в самолете, в формате, описанном выше, причем места, занятые текущей группой пассажиров, должны быть обозначены символом X. Если группа не может найти места, удовлетворяющие ее требованиям, то выведите строку Cannot fulfill passengers requirements.

**Ввод**

4

...\_.#.

.##\_…

.#.\_.##

...\_…

7

2 left aisle

3 right window

2 left window

3 left aisle

1 right window

2 right window

1 right window

**Вывод**

Passengers can take seats: 1B 1C

.XX\_.#.

.##\_...

.#.\_.##

...\_...

Passengers can take seats: 2D 2E 2F

.##\_.#.

.##\_XXX

.#.\_.##

...\_...

Passengers can take seats: 4A 4B

.##\_.#.

.##\_###

.#.\_.##

XX.\_...

Cannot fulfill passengers requirements

Passengers can take seats: 1F

.##\_.#X

.##\_###

.#.\_.##

##.\_...

Passengers can take seats: 4E 4F

.##\_.##

.##\_###

.#.\_.##

##.\_.XX

Cannot fulfill passengers requirements

## Сопроводительное письмо - 10 баллов

 Требуется написать письмо в свободной форме о том, почему Вам интересна робототехника и программирование. Опишите уже имеющийся у Вас опыт в данных областях.

# Приложение №2

#

# Литература и курсы

1. <https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html>
2. <https://habr.com/ru/post/482220/>
3. <https://kpolyakov.spb.ru/uni/teapot.htm>
4. <http://orbicraft3d.sputnix.ru/doku.php>
5. Язык программирования Python <https://www.lektorium.tv/python>

Базовый курс. Вы познакомитесь с типами данных, условиями и циклами в Python, узнаете, как работать со структурами данных, объявлять функции и использовать встроенные методы для решения типовых задач в программировании

1. АЗЫ программирования на Python <https://stepik.org/course/126739/>

Курс для школьников, начиная с 6-7 классов для знакомства с алгоритмами и языком программирования Python.

1. Программирование для школьников на Python. Фарватер <https://stepik.org/course/114239/>
2. "Поколение Python": курс для начинающих <https://stepik.org/course/58852/>
"Поколение Python": курс для продвинутых <https://stepik.org/course/68343/>

Курс с большим количеством тренировочных задач, удобный как для самостоятельного изучения, так и для работы в группе в рамках внеурочной деятельности.

1. Основы программирования на языке Python для школьников <https://stepik.org/course/9232/>

Модуль "Начало" предназначен для школьников 7 - 9 классов и для тех, кто впервые знакомится с языком программирования Python.

Модуль "Вторая ступень. Погружение в Python" предназначен для школьников 9 - 11 классов на базовом и профильном уровнях