

**Задания отборочного тура Открытой Олимпиады СПбГУ среди студентов и молодых специалистов «Petropolitan Science (Re)Search» в 2023/24 учебном году по предмету «Вычислительные технологии» для обучающихся и выпускников бакалавриата**

**Задание 1.** Необходимо написать программу на языке C/C++, которая решает sudoku в общем случае. Запрещено использовать сторонние библиотеки. Напомним, что sudoku - это популярная головоломка, которая состоит из квадратной сетки размером 9x9 клеток, разделенной на девять равных квадратных секций размером 3x3 клетки. В каждой клетке необходимо заполнить цифру от 1 до 9 таким образом, чтобы каждая строка, каждый столбец и каждая секция содержали все цифры от 1 до 9 без повторений. Заметим, что в общем случае, размер поля может быть любой, например, 4x4, 9x9 и т.д.

В качестве ответа необходимо предоставить текст с описанием алгоритма и анализом вариантов решения, а также ссылку на открытый репозиторий GitHub (или аналогичный) с кодом реализованной программы (дата последнего коммита не позже даты завершения отборочного тура олимпиады) или приложите ссылку на архив с кодом. Помните о необходимости соблюдения анонимности при предоставлении материалов.

Входные данные:

Первая строка содержит целое число  $n$ . Далее, если существует поле для sudoku размера  $n \times n$ , идут  $n$  строк, в каждой из которых через пробел записаны  $n$  целых чисел. Вместо пустой клетки поля программа выводит число 0 (для лучшего понимания посмотрите тестовый пример).

Выходные данные:

Если поля для sudoku размером  $n \times n$  не существует, выведите «Invalid field» и завершите программу.

Если поле существует, считайте его, решите задачу и выведите ответ. Ответом для задачи является заполненная матрица размером  $n \times n$ . Каждая новая строка матрицы должна выводиться с новой строки, а числа в каждой строке разделены пробелом. Гарантируется, что для входной матрицы существует решение.

Пример:

Ввод:

```
4
1 4 3 2
0 0 1 4
4 1 2 3
2 3 0 0
```

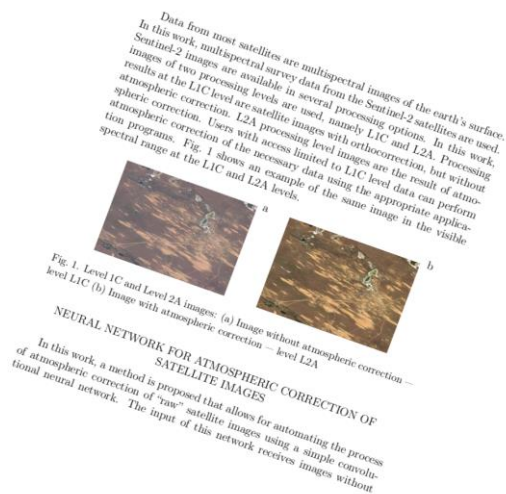
Вывод:

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

**Задание 2.** Найти угол поворота изображения документа, содержащего печатный текст и фото. Скорректировать изображение путем поворота, используя найденный угол. Примеры исходного и повернутого изображения прилагаются.

В качестве ответа необходимо предоставить текст с описанием предлагаемого алгоритма, а также ссылку на открытый репозиторий GitHub (или аналогичный) с кодом реализованной программы (дата последнего

коммита не позже даты завершения отборочного тура олимпиады) либо ссылку на архив с кодом. Помните о необходимости соблюдения анонимности при предоставлении материалов.



Data from most satellites are multispectral images of the earth's surface. In this work, multispectral survey data from the Sentinel-2 satellites are used. Images of two processing levels are available in several processing options. In this work, results at the L1C level are satellite images with orthorectification, but without atmospheric correction. L2A processing level images are the result of atmospheric correction. Users with access limited to L1C level data can perform atmospheric correction of the necessary data using the appropriate application programs. Fig. 1 shows an example of the same image in the visible spectral range at the L1C and L2A levels.

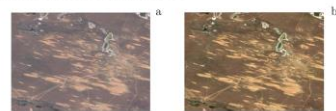


Fig. 1. Level 1C and Level 2A images: (a) Image without atmospheric correction - level 1C (b) Image with atmospheric correction - level 2A

#### NEURAL NETWORK FOR ATMOSPHERIC CORRECTION OF SATELLITE IMAGES

In this work, a method is proposed that allows for automating the process of atmospheric correction of "raw" satellite images using a simple convolutional neural network. The input of this network receives images without

**Задание 3.** Для защиты данных используют разные методы или техники, начиная от простых шифров до гомоморфного шифрования. Многие считают, что для защиты персональных данных хватит и обычных шифров, так как другие методы, такие как AES-128, SHA-512 и т.п. очень сложны или требуют существенных вычислительных ресурсов.

Ваша задача – определить алгоритм, с помощью которого зашифрованы представленные ниже данные; написать программу, предпочтительно используя язык программирования Python, которая деобезличивает этот набор данных.

Почта	Адрес
<a href="mailto:vaeqbt52@symux.oay">vaeqbt52@symux.oay</a>	Бмбшмшъоэцфх ысь.р.37 цо.399
<a href="mailto:zxtwc.hxcrtgt@vbpax.rdb">zxtwc.hxcrtgt@vbpax.rdb</a>	юя. № 4914у.14 щс.284
<a href="mailto:zgzxov.cdgg@ojmkct.xjh">zgzxov.cdgg@ojmkct.xjh</a>	2-ф Дгяегчжяхф иа.щ.73 яч.70
<a href="mailto:yohflz@nsvcly.jvt">yohflz@nsvcly.jvt</a>	87-р су УСЗЛл.16 сй.158
<a href="mailto:wfiu.cvsjrtb@yreu.tfd">wfiu.cvsjrtb@yreu.tfd</a>	1-ъ Васвяюсьщуыяувыщъ ацб.х.35 ыу.167
<a href="mailto:rnpfdqf56@mtyrfnq.htr">rnpfdqf56@mtyrfnq.htr</a>	шр. Межуручбкй.32 пз.476
<a href="mailto:ypobnct.qdnat@vddslxc.dgv">ypobnct.qdnat@vddslxc.dgv</a>	Адзуюфьящчш бвяу.у.37 щс.368
<a href="mailto:rpqnsi@bnxtep.htr">rpqnsi@bnxtep.htr</a>	Бужучузцпед шр.й.57 пз.41
<a href="mailto:puijggck@zglesad.uge">puijggck@zglesad.uge</a>	Баэчфтс еэ.ц.68 ьф.257
<a href="mailto:mzqjmzbw.tqvl@ouiqt.kwu">mzqjmzbw.tqvl@ouiqt.kwu</a>	Иугфцки ыу.м.76 тк.49

В качестве ответа предоставить объяснение способа определения загаданного алгоритма обезличивания данных и назвать сам алгоритм, представить краткое описание реализованного решения, блок-схему алгоритма решения, результат работы программы: деобезличенный датасет с добавлением столбца, в котором указан ключ шифрования; предоставить ссылку на открытый репозиторий GitHub (или аналогичный) с кодом реализованной программы (дата последнего коммита не позже даты завершения отборочного тура олимпиады) либо ссылку на архив с кодом. Помните о необходимости соблюдения анонимности при предоставлении материалов.

### **Критерии оценки решений задач**

**Задание 1.** Общий балл за решение задачи выставляется как сумма баллов за соответствие решения каждому из критериев.

- I. Программа корректно работает для  $n < 9$  (2 балла)
- II. Анализ кода: программа написана понятно и красиво (1 балл)
- III. Медленное решение для любого  $n$  (4 балла)
- IV. Быстрое решение для любого  $n$  (6 баллов)
- V. Рассмотрены возможности распараллеливания алгоритма (описаны изменения алгоритма для применения параллельного выполнения, проанализированы варианты подходящих технологий параллельного программирования, представлена реализация) (4 балла)

**Задание 2.** Общий балл за решение задачи выставляется как сумма баллов за соответствие решения каждому из критериев.

- I. Выбраны подходящие алгоритмы для определения угла и проведения поворота изображения, приведено их описание (5 баллов)
- II. Представлено обоснование предложенного решения с точки зрения работы алгоритмов, занимаемой памяти и используемых ресурсов (4 балла)
- III. Представлен программный код, реализующий алгоритмы (4 балла)
- IV. Проведена экспериментальная оценка точности работы алгоритма с использованием программного решения. Оценка проводится на нескольких изображениях (не менее трех) различных документов, содержащих фото в разных местах документов. (4 балла)

**Задание 3.** Общий балл за решение задачи выставляется как сумма баллов за соответствие решения каждому из критериев.

- I. Правильно определен использованный алгоритм обезличивания (2 балла)
- II. Представлена корректная блок-схема алгоритма деобезличивания (4 балла)
- III. Представлен программный код, реализующий предложенную задачу (5 баллов)
- IV. Программа выводит корректный результат: файл с деобезличенным набором данных (3 балла)
- V. В итоговом файле выведен столбец, где в каждой строке указан верный ключ шифрования (2 балла)

Максимальное количество баллов, которое можно получить за выполнение заданий:

Задание 1 – 17 баллов.

Задание 2 – 17 баллов.

Задание 3 – 16 баллов.

Максимальный балл за всю работу – 50 баллов.