# Вариант 9 конкурсных заданий *практического* этапа Конкурса

В далекой неизученной галактике произошла страшная катастрофа. Все корабли, бороздящие просторы космоса в этой области, затянуло в гигантскую чёрную дыру, искажающую пространство и время. В результате этого, космические судна потеряли свои изначальные координаты, и теперь их судьба была неизвестна. В черной дыре действовали совершенно другие законы физики, и выбраться из нее было практически невозможно. Но команды кораблей не сдавались, они продолжали бороться за свою жизнь, используя последние достижения науки и техники. На одном из кораблей смогли отправить файл с данными о кораблях на Землю и Вы смогли уловить сигнал и получить данные, но из-за далекого расстояния данные немного побились. Разберитесь, что было в исходном файле.

Думаю, что Вы готовы приступать, но сначала необходимо **создать репозиторий** для проекта и задач, которые Вы будете выполнять. Обязательно сделайте его **PUBLIC**, а то мы не сможем проверить решение вашей последней задачи. Каждую задачу вам необходимо будет правильно оформить и залить в репозиторий. *Код, который Вы напишете необходимо задокументировать, чтобы другие программисты могли понять, что делает код и за что отвечает. Примеры документирования приведены ниже.*

**Пример документирования кода на языке С++**

/\*\*

\* Это описание функции foo

\*

\* @param str это описание аргумента str

\* @param pattern это описание аргумента pattern

\* @return это описание того, что вернет функция

\*/

int foo(std::string str, std::string& pattern)

{

...

};

**Пример документирования на языке Python**

def complex(real=0.0, imag=0.0):

"""Описание функции complex.

Описание аргументов:

real – описание аргумента

imag – описание аргумента

"""

if imag == 0.0 and real == 0.0: return complex\_zero

...

Теперь можете приступать к решению задач!

К задачам прикреплен файл *space.csv,* который хранит в себе информацию о имени корабля, название родной планеты корабля, координаты последнего места связи, направление движения.

Столбцы: *ShipName(Код корабля-номер корабля), planet, coord\_place(в формате x, y - целочисленные, через пробел), direction(в формате вектора размерности два через пробел, числа - целочисленные)*

Разделитель *«,».*

*Кодировка: “utf-8”*

***Задача 1.***

Пока мы принимали сигнал возникали помехи и некоторые данные не записались. В столбце *coord\_place* наибольшее число побитых данных, те тех где вместо координат лежат нули. Пожалуйста, составьте отчет для штаба и вышлите его в файле *space\_new.txt.*

**Формат строк в отчете:** “При получении данных о корабле <*ShipName*> возникли сбои. Предположительные координаты - .”

Где , а *N* - количество символов в названии родной планеты, и - координаты направления соответственно.

*В задаче запрещено использование сторонних библиотек(Pandas и др)*

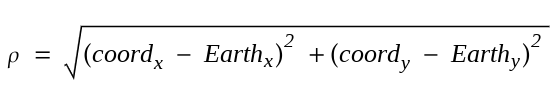
*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка (для языка Python – PEP 257). После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.*

*Решение:*

***Задача 2***

Через какое-то время было принято сложное решение отправить поисковый корабль по координатам самым близким к Земле. Для этого отсортируйте столбец *coord\_place* в порядке возрастания по расстоянию от последнего места связи и Земли. Расчеты производить исходя из того, что координаты Земли: (0, 0).

Формула расчета расстояния:



Также выведите тройку кораблей у которых координаты не побиты, те не равны 0.

**Формат вывода:** “Корабль: <*ShipName*> последний раз был на связи по координатам: <*coord\_place*>; и двигался в направлении: <*direction*>.”

В задаче нельзя использовать сторонние сортировки, для решения воспользуйтесь **быстрой сортировкой**.

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория*

***Задача 3***

**Ввод: стандартный ввод**

**Вывод: стандартный вывод**

Напишите программу, которая по заданному названию корабля выводит его расстояние до Земли по последнему месту связи:

“Корабль <ShipName> последний раз был на связи по координатам: <*coord\_place*>, что составляет: <> космических единиц.”

Формулу для расчета расстояния возьмите из предыдущей задачи. Результаты округлите до 3х знаков после запятой.

Программа должна постоянно самостоятельно запрашивать название корабля(без вопроса) и остановиться только в случае ввода ‘stop’.

В случае, если корабль будет не найден вывести: “error.. er.. ror..”

Поиск необходимо осуществить с помощью линейного алгоритма поиска**.**

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория*

***Задача 4***

Для того, чтобы связаться с экипажами кораблей, было принято решение использовать Азбуку Морзе. Реализуйте методы/функции, которые будут генерировать сообщения для пропавших кораблей. Сообщение должно состоять из кода корабля + его координат.

Азбука Морзе имеет следующие значения:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Русский символ** | **Код Морзе** | **Английский символ** | **Код Морзе** | **Символ** | **Код Морзе** |
| **A** | .- | **A** | **.-** | **1** | .---- |
| **Б** | -... | **B** | **-...** | **2** | ..--- |
| **В** | .-- | **C** | **-.-.** | **3** | ...-- |
| **Г** | --. | **D** | **-..** | **4** | ....- |
| **Д** | -.. | **E** | **.** | **5** | ..... |
| **Е, Ё** | . | **F** | **..-.** | **6** | -.... |
| **Ж** | ...- | **G** | **--.** | **7** | --... |
| **З** | --.. | **H** | **....** | **8** | ---.. |
| **И** | .. | **I** | **..** | **9** | ----. |
| **Й** | .--- | **J** | **.---** | **0** | ----- |
| **К** | -.- | **K** | **-.-** | **Точка** | .-.-.- |
| **Л** | .-.. | **L** | **.-..** | **Запятая** | --..-- |
| **М** | -- | **M** | -- | **Двоеточие** | ---... |
| **H** | -. | **N** | **-.** | **;** | -.-.-. |
| **О** | --- | **O** | **---** | **(** | -.--. |
| **П** | .--. | **P** | **.--.** | **)** | **-.--.-** |
| **P** | .-. | **Q** | **--.-** | **Апостроф** | .----. |
| **С** | ... | **R** | **.-.** | **Кавычки** | .-..-. |
| **Т** | - | **S** | **...** | **-** | -....- |
| **У** | ..- | **T** | **-** | **/** | -..-. |
| **Ф** | ..-. | **U** | **..-** | **?** | ..--.. |
| **Х** | .... | **V** | **...-** | **!** | -.-.-- |
| **Ц** | -.-. | **W** | .-- | **Пробел** | / |
| **Ч** | ---. | **X** | **-..-** |
| **Ш** | ---- | **Y** | -.-- |
| **Щ** | --.- | **Z** | --.. |
| **Ь** | -..- |  |  |
| **Ы** | -.-- |  |  |
| **Ъ** | --.-- |  |  |
| **Э** | ..-.. |  |  |
| **Ю** | ..-- |  |  |
| **Я** | .-.- |  |  |

*Азбука Морзе представлена в файле morse\_code.txt.*

*На вход подается CVS файл, который необходимо записать в список, для каждого кода корабля сгенерировать сообщение, после чего дополнить список сгенерированными элементами. Получившейся результат необходимо записать в новый файл message\_for\_ship.csv.*

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.*

***Задача 5***

Файл "space.csv" содержит информацию о космических кораблях. Ваша цель - создать хеш-таблицу, где ключом будет являться название планеты, а значием количество кораблей, которые были отправлены с этой планеты.

1. Прочитайте данные из файла "space.csv" и загрузите их в память вашей программы.

2. Определите хеш-функцию, которая будет преобразовывать название корабля в индекс хеш-таблицы.

3. Создайте массив (список) для хранения данных о кораблях. В каждом элементе массива будет храниться список (цепочка), который будет использоваться для размещения кораблей с одинаковым хешем.

4. Выведите 5 планет с наибольшим количеством кораблей.

Выведите информацию о корабле, если он был найден, в противном случае, выведите сообщение: "Корабль не найден".

Обратите внимание на производительность вашей хеш-таблицы и ее способность обрабатывать коллизии без потери данных.

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.*

***Задача 6***

Ваш код будет использоваться программистом, которого возьмут на работу, поэтому он должен быть правильно оформлен и выложен на GitHub. Весь написанный код должен быть задокументирован согласно стандартам документирования кода выбранного языка.

Также необходимо оформить README.md для Вашего репозитория. Пункты, которые должны быть описаны:

1. Название проекта

2. Описание проекта

3. Оглавление (необязательно)

4. Как установить и запустить проект

5. Как использовать проект