

Условия

Код решения на языке C# необходимо вписывать в [шаблонный файл](#), не меняя название и структуру уже описанных функций. Можно дописывать в данный файл необходимые вам дополнительные функции, классы/структуры. Весь код должен быть в одном файле - для удобства проверки задания. Код должен компилироваться.

Шаблон решения содержит несколько тестов на задачу. Запуск скачанного шаблона будет писать в консоль ошибки, потому что в методах нет решения, они выдают неправильные ответы. Ваша задача - эти решения написать, чтобы тесты проходили) И запуск скрипта выводил **All Test completed!**

При проверке ваше решение будет запущено на значительно большем количестве тестов, чем в шаблоне. И это будет влиять на оценку вашего тестового задания. Продумайте все узкие места и возможные варианты входных параметров. Гарантируем, что тесты не нарушают описанные ограничения входных параметров.

Дополнительно: будет плюсом чистый код (понятное именование переменных, лаконичная организация кода и тд), оптимальность решение - скорость его работы и потребление памяти.

Задание 1. Слова из слова

Написать функцию, которая для данного слова найдет все слова, которые можно собрать из букв данного слова. Слова вывести отсортированные лексикографически.

keyWord - исходное слово для генерации слов, $2 < \text{длина} \leq 15$
words - список слов, $0 < \text{длина} \leq 15000$

Примеры

слово: "арбуз"
словарь: {"бра", "раб", "зубр", "кот", "ток"}
результат: {"бра", "зубр", "раб"}

слово: "мама"
словарь: {"кот", "ток", "мимо"}
результат: {} - из букв заданного слова нельзя собрать ни одно слово из словаря, результат пустой массив

Задание 2. Два уникальных символа

Исходная строка состоит из маленьких латинских букв [a-z]. Составим альтернативную строку по следующему алгоритму

- будем удалять из строки по символу пока в строке не останется ровно два вида уникальных символа (например, только символы a и b)

- если мы удаляем символ - то мы удаляем все его вхождения в строку (если убираем 'а', то все символы 'а' удаляются из строки)

- в получившейся строке символы не должны повторяться подряд (например, строка aab - неверная, символ 'а' повторяется подряд два раза)

Вывести максимальную длину правильной альтернативной строки, которую можно получить такой трансформацией. Если получить валидную строку нельзя, то вывести 0.

Ограничения

1 <= длина строки <= 1000

Примеры

строка: beabeeab
результат: 5 //babab

строка: a
результат: 0

строка: ab
результат: 2

Пояснение: удаляем символ 'е' - получаем строку, состоящую из двух уникальных символов а и b максимальной длины для данной строки.

Задание 3. Предыдущее число

Дано число, составить из тех же цифр данного числа предыдущее число - максимальное из меньших чисел составленных из цифр данного числа. Если такое число нельзя составить, вывести -1. Если из цифр текущего числа можно составить меньшее число, но с ведущим 0, то - также вывести -1.

Ввод

0 <= число <= 1,000,000,000,000

Вывод

0 <= число <= 1,000,000,000,000

Примеры

число: 21
результат: 12

число: 531
результат: 513

число: 1027
результат: -1 // 0721 не подходит так как получается ведущий ноль

Задание 4. Конь и Королева

Дана клеточная матрица $N \times M$, есть две шахматные фигуры конь и королева. Найти количество ходов, за которые каждая фигура дойдет до заданной конечной клетки из заданной начальной. Клетки, в которые можно ходить, помечены '#', клетки стены отмечены 'x'. Фигуры во время своих ходов не пересекаются/не влияют друг на друга. Вывести массив из двух чисел - [ходы_коня, ходы_королевы]. Если цель не достижима, вывести -1. Начальная клетка помечена символом s, конечная помечена символом e

- фигуры ходят согласно шахматным правилам (см. википедию)
- фигура не может становиться на стену
- королева не может проходить сквозь стены
- конь может перепрыгивать через стены и становиться на свободную клетку

Ограничения

$$2 \leq N, M \leq 1000$$

Пример

```
s # # # #
# x x x x #
# # # # x #
# # # # x #
# # # # e
```

результат: {3, 2}

```
s # # # # x
# x x x x #
# x # # x #
# # # # x #
x # # # e
```

результат: {-1, 3}

```
s # # # # x
x x x x x
# # # # x #
# # # e x #
x # # # #
```

результат: {2, -1}

Задание 5. Жадина

В игре есть города связанные между собой дорогами, на которых разбросаны монетки. Есть начальная и конечная точки (индексы городов). Игрок должен пройти из заданного города А в заданный город Б. Посещать каждый город можно только один раз. Найти максимальное количество монет, которые может

собрать игрок пройдя из начальной точки в конечную. Карта задается набором дорог, города нумеруются с 0. У дорог нет направлений - если из А можно пройти в Б, то из Б можно пройти по этой дороге в А. Если до конечной точки нельзя добраться то вывести -1.

Ограничения

2 <= количество городов <= 100

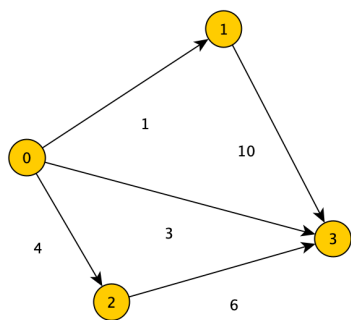
1 <= количество монет дороге <= 10000

Пример

ввод карты

```
0 1 1
0 2 4
0 3 3
1 3 10
2 3 6
```

//где первое и второе числа индексы городов соединенных дорогой, третье число кол-во монет на дороге



старт: 0

финиш: 3

результат: 11

Важно

Обрати внимание, что задания необходимо выполнить в шаблонном файле, который можно скачать [здесь](#). Чтобы техническое задание было принято, прикрепи файл в формате cs в специально отведенное поле в опроснике google форме (решение всех задач должно находиться в одном файле).