Алгоритмика для развлечений

Рекомендация к выполнению

- помним про оформление кода;
- хотелось бы посоветовать как можно большее использование стандартной библиотеки C++, в том числе – контейнерных типов (**vector, map, set, list, stack, queue**) и всяких прикольных штук из **<algorithm>**. Да и про **string** не забываем.

Задания

4.1. Перевести заданное целое число в римскую систему счисления.

Символ	Число
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

В целом, римские числа записываются начиная с больших символов к меньшим. Так, число XXXVIII можно расписать как XXX+V+III, что при переводе даёт $30+5+3=38$. Но имеются следующие исключения:

- символ «I» может стоять перед «V» и «X» для обозначения чисел 4 («IV») и 9 («IX»);
- символ «X» может стоять перед «L» и «C» для обозначения чисел 40 («XL») и 90 («XC»);
- символ «C» может стоять перед «D» и «M» для обозначения чисел 400 («CD») и 900 («CM»).

Ограничение на заданное для перевода число – 3999, поскольку далее в римской системе счисления начинаются обозначения, которые довольно сложно выразить в ASCII-кодировке [1].

4.2. Подстрока максимальной длины.

Найти в произвольной строке максимальную длину **подстроки**, в которой нет ни одного повторяющегося символа.

Примеры.

```
Входной параметр: "trttwe6bm6ad"
Ответ:            "twe6bm"

Входной параметр: "wwww"
Ответ:            "w"

Входной параметр: "wdaakva2"
Ответ (один из):  "wda" или "akv"
```

[1] пример тут и тут

4.3.Преобразование строки под кодовой кличкой «зигзаг». Для входной строки и заданного *целого* числа, представляющего условное количество строк, разбить исходную строку и считать новую строку по следующему примеру:

```
Входные параметры: "stopcrazyworld", 4
Ответ:             "saltrzrdocyopw"

Как получилось:
s   a   l
t  rz  rd
o c y o
p   w
```

Заполнили зигзаг в соответствии с заданным количество строк и считали все непустые символы с первой по последную строки.

4.4. Найти самый длинный палиндром во входящей строке

Задана произвольная строка. Найти в ней подстроку **наибольшей длины**, которая является палиндромом.

```
Входной параметр: "wghgtyk"
Ответ:            "ghg"

Входной параметр: "wtcr6uaba"
Ответ:            "wtc" или "aba"

Входной параметр: "kvaawk"
Ответ (один из):  "aa"
```

4.5. Удалить дубликаты из отсортированного массива.

Дан массив целых чисел, отсортированных по возрастанию. Удалить из него все дубликаты и **модифицировать его** так, чтобы первые k-мест занимали уникальные числа.

Ограничение: не использовать дополнительные массивы (т.е., не разрешено никакое дополнительное выделение памяти).

4.6. Поиск значения в отсортированном, но повёрнутом, массиве

Дан отсортированный по возрастанию массив целых чисел. Он может быть **повёрнут** на произвольное число k такми образом, что его элементы расположились следующим образом:

```
[
  numbers[k], numbers[k + 1], ... , numbers[len - 1],
  numbers[0], ... , numbers[k-1]
]
```

Задано число target для поиска в массиве. Найти его позицию (или вернуть «-1», если не найдена) но с условием: прямой перебор по элементам массива запрещён. Другими словами, алгоритм должен работать быстрее (в среднем), чем $\mathrm{O(len)}$, где len – количество элементов массива.