

Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники

## Алгоритмы и структуры данных

Отчёт по лабораторной работе №7 (1322)

Преподаватель: Тропченко А. А.

Выполнил: Марухленко Д. С.

Группа: R3235

Санкт Петербург, 2021г.

## 1. Цель работы

Решить задачу №1322 на платформе Timus Online Judge

<https://acm.timus.ru/problem.aspx?space=1&num=1322>

## 2. Задача

### Условие

Спецслужбы обнаружили действующего иностранного агента. Шпиона то есть. Установили наблюдение и выяснили, что каждую неделю он через Интернет посылает кому-то странные нечитаемые тексты. Чтобы выяснить, к какой информации получил доступ шпион, требуется расшифровать информацию. Сотрудники спецслужб проникли в квартиру разведчика, изучили шифрующее устройство и выяснили принцип его работы.

На вход устройства подается строка текста  $S_1 = s_1s_2\dots s_N$ . Получив ее, устройство строит все циклические перестановки этой строки, то есть  $S_2 = s_2s_3\dots s_Ns_1$ , ...,  $S_N = s_Ns_1s_2\dots s_{N-1}$ . Затем множество строк  $S_1, S_2, \dots, S_N$  сортируется лексикографически по возрастанию. И в этом порядке строчки выписываются в столбец, одна под другой. Получается таблица размером  $N \times N$ . В какой-то строке  $K$  этой таблицы находится исходное слово. Номер этой строки вместе с последним столбцом устройство и выдает на выход.

Например, если исходное слово  $S_1 = \text{abracadabra}$ , то таблица имеет такой вид:

1. aabracadabr =  $S_{11}$
2. abraabracad =  $S_8$
3. abracadabra =  $S_1$
4. acadabraabr =  $S_4$
5. adabraabrac =  $S_6$
6. braabracada =  $S_9$
7. bracadabraa =  $S_2$
8. cadabraabra =  $S_5$
9. dabraabraca =  $S_7$
10. raabracadab =  $S_{10}$
11. racadabraab =  $S_3$

И результатом работы устройства является число 3 и строка rdarcaaaabb.

Это все, что известно про шифрующее устройство. А вот дешифрующего устройства не нашли. Но поскольку заведомо известно, что декодировать информацию можно (а иначе зачем же ее передавать?), Вам предложили помочь в борьбе с хищениями секретов и придумать алгоритм для дешифровки сообщений. А заодно и реализовать дешифратор.

Ограничение времени: 0.25 секунды

Ограничение памяти: 64 МБ

### Исходные данные

В первой и второй строках находятся соответственно целое число и строка, возвращаемые шифратором. Длина строки и число не превосходят 100000. Строка содержит лишь следующие символы: a-z, A-Z, символ подчеркивания. Других символов в строке нет. Лексикографический порядок на множестве слов задается таким порядком символов:

ABCDEFGHIJKLMNOPQRSTUVWXYZ\_abcdefghijklmnopqrstuvwxyz

Символы здесь выписаны в порядке возрастания.

### Результат

Выведите декодированное сообщение в единственной строке.

### Пример

Исходные данные	Результат
3 rdarcaaaabb	abracadabra

## 3. Материалы работы

### 3.1. Объяснение алгоритма

Для решения задачи считаем строку посимвольно в массив структур, содержащих букву и порядковый номер считанной буквы. После этого отсортируем по алфавиту, сохраняя порядок повторяющихся букв функцией `stable_sort`. Полученная строка совпадает со строкой, полученной перестановками и сортировкой исходной строки (в примере S11), а ряд индексов указывает на последовательность расстановки букв и место индекса следующей буквы. Таким образом, достаточно один раз пропустить структуру через цикл, чтобы получить исходное слово.

### 3.2. Код программы.

```
1. #include <algorithm>
2. #include <iostream>
3.
4. using namespace std;
5.
6. struct char_i {
7.     char c;
8.     int index;
9. };
10.
11. bool sort_char_i(const char_i a, const char_i b) {return a.c < b.c;}
12.
13. int main() {
14.     int n;
15.     cin >> n;
16.     --n;
17.     char c;
18.     char_i s[100000];
19.     int length = 0;
20.     while (cin.get(c)) {
21.         if (length > 0 && (c == '\n' || c == '\r'))
22.             break;
```

```

23.         else if (isspace(c))
24.             continue;
25.         s[length] = {c, length};
26.         ++length;
27.     }
28.
29.     stable_sort(s, s+length, sort_char_i);
30.
31.     int j = n;
32.     for (int i = 0; i < length; ++i, j = s[j].index)
33.         cout << s[j].c;
34.     cout << endl;
35.
36.     return 0;
37. }

```

#### 4. Результат выполнения и ссылка на репозиторий GitHub

задача: [ШПИОН](#)

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
9259155	01:37:43 9 мар 2021	<a href="#">Daniil Marukhlenko</a>	<a href="#">1322</a>	G++ 9.2 x64	Accepted		0.015	1 564 КБ

[https://github.com/japersik/algorithms\\_and\\_data\\_structures/](https://github.com/japersik/algorithms_and_data_structures/)



#### 5. Вывод

Работа выполнена, задача решена с использованием сортировки структур.