

*Надточий Александра Михайловна
Ульянцев Владимир Игоревич,
Цыпленков Алексей Евгеньевич*

ЗАДАЧА «КОНСПИРАЦИЯ»

Этой статьей мы начинаем очередной цикл публикаций олимпиадных задач для школьников по информатике и программированию с разборами. Решение таких задач и изучение разборов поможет Вам повысить уровень практических навыков программирования и подготовиться к олимпиадам по информатике и программированию.

В этой статье рассматривается задача «Конспирация», которая предлагалась в четвертой интернет-олимпиаде базового уровня сезона 2011–2012. Интернет-олимпиады по информатике базового уровня проводятся Санкт-Петербургским национальным исследовательским университетом информационных технологий, механики и оптики. Сайт этих олимпиад находится по адресу <http://neerc.ifmo.ru/school/io/>.

УСЛОВИЕ ЗАДАЧИ

Петя и Вася играют в шпионов. Главная цель этой игры – обмениваться секретными сообщениями, сохраняя конспирацию. Для этого Петя и Вася используют особый шифр.

Метод шифровки и дешифровки очень прост: каждой букве α русского алфавита сопоставляется некоторая последовательность букв латинского алфавита β . Затем в сообщении все вхождения буквы α заменяются на соответствующую последователь-

ность β . А чтобы совсем запутать вероятного противника, Петя и Вася договорились, что значимым в исходном сообщении должно быть только лексикографически минимальное слово. Напомним, что строка a лексикографически меньше строки b , если существует такое i , что для любого $j < i$ выполняется равенство $a_j = b_j$ и при этом $a_i < b_i$, или строка a является префиксом строки b .

Недавно Петя предложил Васе использовать следующую таблицу преобразования букв (табл. 1).

Однако теперь Вася не успевает дешифровать сообщения Пети вовремя и просит вас помочь ему – определить номер слова, которое до шифрования, записанное русски-



Табл. 1

а	a	з	z	п	p	ч	ch
б	b	и	i	р	r	ш	sch
в	v	й	j	с	s	щ	sesh
г	g	к	k	т	t	ы	yy
д	d	л	l	у	u	э	ye
е	e	м	m	ф	f	ю	yu
ё	yo	н	n	х	kh	я	ya
ж	zh	о	o	ц	tc		

ми буквами, было лексикографически минимальным.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \leq n \leq 1000$) – количество слов в шифровке. Следующие n строк содержат по одному слову, состоящему из маленьких латинских букв – текст шифровки, которую получил Вася. Длина каждого слова не превышает 50 символов.

Формат выходного файла

В выходной файл выведите номер слова шифровки, которое до шифрования было лексикографически минимальным в русском алфавите. Шифровки нумеруются с единицы в порядке появления во входном файле.

Примеры входных и выходных данных

conspiracy.in	conspiracy.out
4 yolka jeti yabloko shtab	1

Примечание

Пояснение к примеру: исходное сообщение состояло из слов «ёлка», «йети», «яблоко», «штаб».

РАЗБОР ЗАДАЧИ

Данная задача разбивается на две подзадачи: дешифровать слово обратно в русский

алфавит и отсортировать слова в лексикографическом порядке. Заметим, что заданная в условии таблица преобразования позволяет однозначно дешифровать слово.

При дешифровке слова следует учитывать, что недостаточно смотреть на один латинский символ и тут же переводить его в русский аналог. Например, если рассматривается символ «z», то, в зависимости от следующего символа (а может быть, это последний символ), можно получить как букву «з», так и букву «ж». Фрагменты процедуры, выполняющей дешифровку, приведены в листинге 1. Листинги содержат код, написанный на языке Pascal.

Исходя из ограничений задачи, возможна реализация любой сортировки, в том числе и сортировки «пузырьком». В данном случае рассмотрим так называемую «быструю сортировку» (*Quicksort*), работающую в среднем за $O(n \log n)$. Фрагмент программы, реализующей быструю сортировку, приведен в листинге 2. Подробное описание алгоритмов сортировки и оценки их времени работы можно найти, например, в [1, 2].

В листинге 3 приведен основной текст программы, использующий описанные функции. По условию задачи требуется вывести номер слова шифровки (а не само слово), поэтому создадим специальную структуру **t**, в которой будем хранить дешифрованное слово вместе с его номером во входном файле. Заметим, что при реализации алгоритма

Листинг 1. Фрагмент программы, реализующий дешифровку слова

```

const // символы, занимающие в латинском написании 1, 2, 3, 4
        СИМВОЛОВ СООТВЕТСТВЕННО
    set1 : set of char = ['a', 'б', 'в', 'г', 'д', 'е', 'з', 'и', 'й',
                        'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф'];
    set2 : set of char = ['ё', 'ж', 'ц', 'ы', 'э', 'ю', 'я', 'х'];
    set3 : set of char = ['ч', 'ш'];
    set4 : set of char = ['щ'];
function russian(s : string) : string;
var
    i : longint;
    ans : string;
begin
    ans := '';
    i := 1;
    while (i <= length(s)) do begin
        if (s[i] = 'a') then ans := ans + 'a';
        if (s[i] = 'b') then ans := ans + 'б';
        ... // аналогично обрабатываются все символы, транслируемые
        одной буквой
        // обработка случаев для букв з и ж
        if (s[i] = 'z') then begin
            if (i = length(s)) then ans := ans + 'з' else
            if ((s[i + 1]) = 'h') then ans := ans + 'ж'
            else ans := ans + 'з';
        end;
        ... // обработка случаев для букв к и х, т и ц производится
        аналогично
        // обработка случаев для гласных е, ё, ю, я, ы
        if (s[i] = 'y') then begin
            if (s[i + 1] = 'o') then ans := ans + 'ё';
            if (s[i + 1] = 'y') then ans := ans + 'ы';
            if (s[i + 1] = 'e') then ans := ans + 'э';
            if (s[i + 1] = 'u') then ans := ans + 'ю';
            if (s[i + 1] = 'a') then ans := ans + 'я';
        end;
        // обработка случаев для букв с, ш, щ
        if (s[i] = 's') then begin
            if (i = length(s)) then ans := ans + 'с'
            else
                if (s[i + 1] <> 'c') then ans := ans + 'с'
                else begin
                    if (s[i + 2] = 'h') then ans := ans + 'ш'
                    else ans := ans + 'щ';
                end;
            end;
        end;
        // увеличиваем i на количество просмотренных СИМВОЛОВ
        if (ans[length(ans)] in set1) then i := i + 1;
        if (ans[length(ans)] in set2) then i := i + 2;
        if (ans[length(ans)] in set3) then i := i + 3;
        if (ans[length(ans)] in set4) then i := i + 4;
    end;
    russian := ans;
end;

```

Листинг 2. Фрагмент программы, реализующий быструю сортировку

```
procedure sort(l, r : longint);
var
  i, j : longint;
  x : string;
  y : t;
begin
  i := l;
  j := r;
  x := a[(l + r) div 2].s; // выбор опорного элемента
  repeat
    while (a[i].s < x) do
      inc(i);
    while (a[j].s > x) do
      dec(j);
    if (i <= j) then begin
      y := a[i];
      a[i] := a[j];
      a[j] := y;
      inc(i);
      dec(j);
    end;
  until (i > j);
  // применяем сортировку к левой и правой частям массива
  if i < r then
    sort(i,r);
  if j > l then
    sort(l,j);
end;
```

Листинг 3. Основной текст программы

```
type
  t = record // структура для хранения шифровки и ее номера
    s : string;
    i : longint;
  end;
var
  i, n, j : longint;
  s : array [0..2000] of string; // исходный массив слов
  a : array [0..2000] of t; // массив дешифрованных слов
begin
  reset(input, 'conspiracy.in');
  rewrite(output, 'conspiracy.out');
  readln(n);
  for i := 1 to n do begin
    readln(s[i]);
    a[i].s := russian(s[i]);
    a[i].i := i;
  end;
  sort(1, n);
  writeln(a[1].i);
end.
```

быстрой сортировки, приведенной в листинге 2, сравнение элементов массива **a** элемен-

тов типа **t** производилось по полю, хранящему слово (к примеру, **a[i].s**).

Литература

1. Кнут Д. Искусство программирования. Том 3. Сортировка и поиск. М.: Вильямс, 2009.
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. М.: Вильямс, 2007.

*Надточий Александра Михайловна,
студентка пятого курса кафедры
«Компьютерные технологии»
НИУ ИТМО,*

*Ульянцев Владимир Игоревич,
студент пятого курса кафедры
«Компьютерные технологии»
НИУ ИТМО, член жюри Интернет-
олимпиад по информатике
базового уровня,*

*Цыпленков Алексей Евгеньевич,
студент второго курса кафедры
«Компьютерные технологии»
НИУ ИТМО, член жюри Интернет-
олимпиад по информатике
базового уровня.*



Наши авторы, 2012.

Our authors, 2012.