

# Разбор задачи «Контрольная по ударениям», лига А

Автор задачи и разбора — Т. Кривошеева

Данную задачу можно отнести к категории "работа со строками и текстом". Выделим основные моменты, создававшие трудности у участников:

## 1. Чтение и хранение словаря.

Под «словом без ударений» будем понимать слово, полученное из данного изменением всех больших букв на такие же маленькие.

При работе со словарем нам потребуется отвечать на два вопроса:

- Встречается ли в словаре такое же слово, как и данное
- Встречается ли в словаре такое же по написанию слово, как и данное, возможно, с другими ударениями (то есть, совпадающие, если не различать большие и маленькие буквы)

Для этих целей предлагается хранить два массива, первый  $a$  — всевозможные слова без ударений, второй ( $b$ ) сам в свою очередь массив [1..30], в  $b[i][j]$ -ячейке хранится 1, если в словаре есть слово  $a[i]$ , в котором стоит ударение на  $j$ -ю букву, и 0 в противном случае:

```
var a : array[0..20000] of string[30];
    b : array[1..20000, 1..30] of integer;
...
m := 0;
fillchar(b, sizeof(b), 0);
for i:=1 to n do begin
  readln(s);
  j := 1; t := 0;
  while (s[j] in ['a'..'z']) do inc(j);
  s := lowercase(s);
  if (m = 0) or (s <> a[m]) then begin
    inc(m);
    a[m] := s;
  end;
  b[m][j] := 1;
end;
```

Каждое слово без ударений запишется в  $a$  ровно один раз, поскольку словарные слова перечислены в алфавитном порядке. Более того, массив  $a$  окажется лексикографически упорядоченным по возрастанию.

## 2. Чтение и обработка текста.

Не требуется хранить весь текст сразу, можно на каждом шаге читать новое слово (оно заканчивается пробелом или концом файла и содержит не более 30 символов). Прочитав, надо проверить количество ударений (то есть заглавных букв). Если оно не равно одному, то это слово сразу засчитывается как ошибка. В противном случае начинаем работу со словарем (см. следующий пункт). Ниже приведен фрагмент программы, читающей слово из маленьких и заглавных латинских букв, заканчивающегося пробелом или концом файла:

```
var s : string;
    c : char;
...
s := '';
while (true) do begin
    read(c);
    if (c in ['A'..'Z', 'a'..'z']) then s := s + c else break;
end;
```

## 3. Поиск слова из текста в словаре.

Остался последний пункт — как определить по прочитанному слову  $s$ , есть ли в нем ошибки.

Так как ограничения на количество слов в словаре достаточно внушительные, искать  $s$  простым перебором будет слишком долго. Переведем данное слово в нижний регистр и отдельно запомним, на какую позицию  $j$  в этом слове падает ударение. Поищем  $lowercase(s)$  в  $a$  бинарным поиском (как было сказано выше, массив  $a$  упорядочен по возрастанию, так что это действительно можно сделать). Если результат отрицательный, ошибки нет. В противном случае помотрим, стояло ли в словаре ударение на ту же позицию:  $b[i][j] = 1$  если да, 0 если нет — только в этом случае имеет место ошибка.

Подсчитав количество ошибок, получим ответ.