

Разбор задачи «D. Смайлики»

Автор задачи — В. Гуровиц, разбора — Б. Василевский

В этой задаче нужно было аккуратно запрограммировать распознавание в тексте. Будем просматривать его слева направо. Заведем переменную s . Она равна 1, если предыдущий символ равен «:» или «;»; $s = 2$ если несколько предыдущих символов равны «-», а символ перед ними равен «:». В противном случае $s = 0$. Тогда разбор происходит очень просто.

- Вначале $s = 0$.
- Если текущий символ равен одной из скобок, то в случае $s = 1$ или 2 нашелся смайлик. Обнуляем s .
- Если текущий символ равен «-», то в случае $s = 1$ нужно увеличить s на единицу.
- Если текущий символ равен «:» или «;», то присвоить s единицу.
- Если текущий символ не равен ни одному из перечисленных, обнуляем s .

Очевидно, что ни один из смайликов не будет пропущен с помощью этого прохода. Далее, поскольку соседние смайлики не пересекаются друг с другом, каждый из них будет посчитан ровно один раз.

Имеется еще одно решение этой задачи. Сначала считаем строку, пропуская символ «-»:

```
var c : char;
    s : string;
    i, a : longint;
begin
  reset ( input, 'd.in' );
  rewrite ( output, 'd.out' );

  s := '';
  while ( not eof and not eoln ) do begin
    read ( c );
    if ( c <> '-' ) then s := s + c;
  end;
```

Теперь достаточно поискать в строке s построки вида $:*$ и $;*$, где $*$ - одна из скобок:

```
a := 0;
for i := 2 to length ( s ) do
  if ( s[i - 1] = ':' ) or ( s[i - 1] = ';' ) then
    if ( s[i] = ')' ) or ( s[i] = '(' ) or ( s[i] = '[' ) or ( s[i] = ']' ) then
      a := a + 1;
write ( a );
close ( input );
close ( output );
end.
```