

## Построение лексического анализатора с использованием lex

### Цели

- Познакомиться с «регулярными выражениями»
- Познакомиться с программой lex
- Научиться строить лексические анализаторы с использованием lex

### Порядок выполнения

1. Построить регулярное выражение
2. Составить файл для lex
3. Получить программу на Си
4. Откомпилировать и запустить
5. Написать отчет

### Состав отчета

- Титульный лист (фамилия, группа, номер варианта, наименование работы, задание)
- Текст задания
- Текст программы на lex
- Результаты работы программы

**Варианты заданий** В задании указано содержательное описание грамматики и простейши пример для облегчения понимания.

1. Одинаковые символы стоят парами: *aabbaabbbbaa*
2. В начале строки *a*: *ababbbabb*
3. Первый символ — не важен, далее одни *a*: *baaaaaaaaaa, aaaaaaaaaa*
4. Либо одни *a*, либо одни *b*: *aaaaaaaaaa, bbbbbbbb*
5. В конце строки *b*: *ababbbabb*

6. Одинаковые символы не должны стоять рядом: *ababababab, bababababa*
7. В строке должна встретиться хотя бы одна буква *a*: *bbbbbabbb, aaaaaaaaaa*
8. Предпоследним символом строки должна быть *b*, *abbabaabb*
9. Вторым символом строки должна быть *a*: *baaaaaabb*
10. Два последних символа должны быть *b*: *abababb, bbbbbb*
11. Первый и третий символы должны быть разными: *aabbbabab, baaabbbab*
12. Первый и последний символы должны быть одинаковыми: *ababababa, babbabab*

## Пример

**Задание** Символы *a* и *b* стоят парами: *abbaabab, baabbaba*.

**Регулярное выражение** Регулярное выражение будет иметь вид:  
 $(ab|ba)^+$

**Программа на lex** Создадим файл *lab3.l* со следующим содержимым:

---

```
%%  
(ab|ba)+ { printf("Yes");}  
.* { printf("No"); }  
%%  
  
yyerror(char *str)  
{ printf(str); }  
  
main()  
{ yylex(); }
```

---

Listing 1: *lab3.l*

**Получение программы на c** Программу на *c* получим простой командой. *lab.c* - имя выходного файла, *lab3.l* имя входного файла на *lex*.

---

```
flex -o lab3.c lab3.l
```

---

Listing 2: трансляция

**Компиляция и запуск** Для компиляции воспользуемся компилятором gcc.

---

```
gcc -o lab3 lab3.c -lfl  
./lab3
```

```
ababbababa
```

```
Yes
```

```
bbbaa
```

```
No
```

---

Listing 3: компиляция и запуск