Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н. Э. Баумана

(национальный исследовательский университет)» (МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Космический факультет» КАФЕДРА «Прикладная математика, информатика и вычислительная техника»

## ОТЧЕТ

по лабораторной работе № 7

по курсу «Системное программное обеспечение» на тему: «Ознакомление с генератором программ лексического разбора LEX»

Вариант № 14

Студент К3-56Б

(Группа) (Подпись, дата)

Преподаватель

(Подпись, дата)

Несмеянов С. А.

(И. О. Фамилия)

Чернышов А.В.

(И. О. Фамилия)

# Задание



# Описание команд

**%%** — раздел, который отделяет определения и правила (грамматику) от основной части кода. Это ключевая часть синтаксиса LEX, которая указывает на начало блока правил.

**[ \t]+** — регулярное выражение, которое определяет пробелы и табуляции. Эта конструкция игнорируется, то есть ничего не происходит, когда встречается один или несколько пробелов. Это нужно для того, чтобы пропустить лишние пробелы между предложениями, не захватывая их в качестве текста.

**([^.?!]+[.!?])** — регулярное выражение для захвата предложений, которое ищет последовательности символов, не содержащие знаков конца предложения (точка, восклицательный знак или вопросительный знак). После того как LEX находит такое выражение, он распознает его как отдельное правило, и выполняется соответствующий блок кода в фигурных скобках.

**while (isspace(\*yytext)) { yytext++; }** — цикл, который пропускает все пробелы в начале захваченного предложения, используя функцию isspace из библиотеки ctype.h. Этот код помогает удалить лишние пробелы, которые могут находиться перед первым словом в предложении, гарантируя, что текст передается в process\_sentence без ведущих пробелов.

**if (isupper((unsigned char)yytext[0])) { process\_sentence(yytext); }** — условие, проверяющее, начинается ли предложение с заглавной буквы. Функция isupper проверяет, является ли первый символ предложения заглавным. Если условие истинно, вызывается функция process\_sentence, которая обрабатывает и выводит предложение. Это гарантирует, что программа будет обрабатывать только корректные предложения, начинающиеся с заглавной буквы, игнорируя все остальные (если предложение начинается со знака препинания или строчной буквы).

**void process\_sentence(char \*sentence)** — определение функции, принимающей указатель на строку (предложение) и выводящей его на экран с помощью printf. Эта функция предназначена для дальнейшей обработки предложений, включая их форматирование или анализ, но в данном коде она просто выводит текст на экран.

**int main(int argc, char \*\*argv)** — основная функция программы, где начинается выполнение. Она принимает аргументы командной строки, которые могут использоваться для указания файла, который необходимо обработать.

**FILE \*file = fopen(argv[1], "r");** — открывает файл для чтения, используя первый аргумент командной строки. Если файл не может быть открыт (например, если он не существует или нет прав доступа), программа сообщает об ошибке с помощью perror.

**yyin = file;** — присваивает указатель на открытый файл yyin, который является стандартным входом для лексического анализатора LEX. Это позволяет LEX обрабатывать текст из файла так же, как если бы он читал его из стандартного ввода.

**yylex();** — вызывается функция лексического анализатора, которая выполняет обработку входного текста. Она будет повторно вызываться до тех пор, пока не достигнет конца файла. Каждое распознаваемое предложение передается в process\_sentence.

# Код программы

%{

#include <stdio.h>

#include <ctype.h>

void process\_sentence(char \*sentence);

%}

%%

[ \t]+ ;

([^.?!]+[.!?]) {

while (isspace(\*yytext)) {

yytext++;

}

if (isupper((unsigned char)yytext[0])) {

process\_sentence(yytext);

}

}

\n ;

%%

void process\_sentence(char \*sentence) {

printf("%s\n", sentence);

}

int main(int argc, char \*\*argv) {

if (argc > 1) {

FILE \*file = fopen(argv[1], "r");

if (!file) {

perror("Error opening file");

return 1;

}

yyin = file;

}

yylex();

return 0;

}

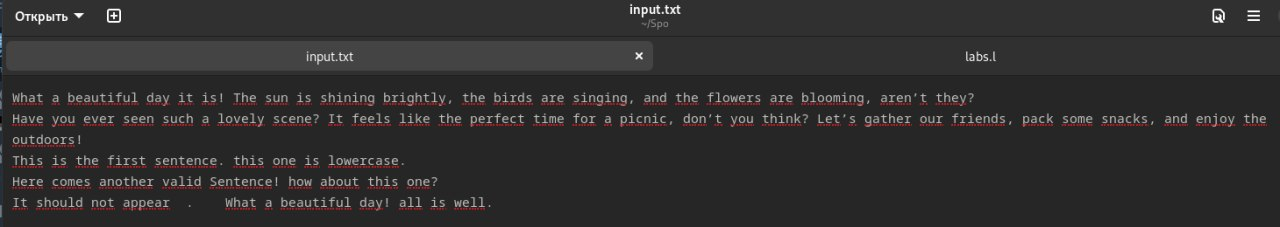
**Сборка и отладка программы:**

flex labs.l

gcc lex.yy.c -lfl -o labs

./labs input.txt

**Текст в файле input.txt**



# Результаты работы программы

# 