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 № 8

по курсу «Системное программное обеспечение»

на тему: «Совместное использование YACC и LEX при помощи утилиты Make»

Вариант № 14

Студент \_\_К3-56Б\_\_  **\_\_\_\_\_\_\_\_\_\_** Несмеянов С. А.

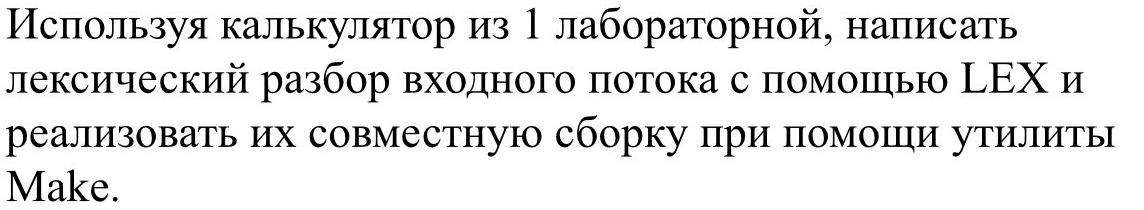
(Группа) (Подпись, дата) (И.О.Фамилия)

#### Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_ Чернышов А. В.

(Подпись, дата) (И.О.Фамилия)

2024 г.

**Задание**



**Список используемых команд и правил**

### В файле Calc.l:

1. **[0-9]+(\.[0-9]+)?** — регулярное выражение, которое определяет числа (целые и с плавающей точкой). При нахождении числа оно преобразуется в double и возвращается как NUMBER.
2. **[ \t]** — пропускает пробелы и табуляции.
3. **\n** — возвращает символ новой строки '\n' для обработки в Calc.y.
4. **.** — возвращает любой другой одиночный символ как есть.
5. **int yywrap(void)** — завершает обработку, возвращая 1, что сообщает Flex об окончании ввода.

### В файле Calc.y:

1. **#include <stdio.h>**, **#include <stdlib.h>**, **#include <math.h>** — подключение библиотек для ввода-вывода, работы с математическими функциями и преобразованием типов.
2. **extern int yylex(void);** — объявление функции лексера yylex, определенной в Calc.l.
3. **void yyerror(const char \*s);** — функция для обработки ошибок.
4. **%union { double dval; }** — объявление объединения для хранения значений.
5. **%token <dval> NUMBER** — объявление токена NUMBER с типом double.
6. **%type <dval> expression** — определение типа для нетерминала expression.
7. **%left '+' '-', %left '\*' '/', %right '^'** — задание приоритетов и ассоциативности операций.
8. **Правила грамматики:**
   * input — принимает выражения и печатает результат.
   * expression — определяет арифметические операции:
     + **NUMBER** — присваивает значению выражения NUMBER.
     + **expression '+' expression** — сложение.
     + **expression '-' expression** — вычитание.
     + **expression '\*' expression** — умножение.
     + **expression '/' expression** — деление.
     + **expression '^' expression** — возведение в степень, использует pow().
     + **'(' expression ')'** — обработка выражений в скобках.
9. **void yyerror(const char \*s)** — выводит сообщение об ошибке.
10. **int main(void)** — основная функция, которая вызывает парсер yyparse.

**Код программы**

**Calc.l**

%{

#include "calc.tab.h"

#include <stdlib.h>

%}

%%

[0-9]+(\.[0-9]+)? { yylval.dval = atof(yytext); return NUMBER; }

[ \t] ;

\n { return '\n'; }

. { return yytext[0]; }

%%

int yywrap(void) {

return 1;

}

**Calc.y**

%{

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

extern int yylex(void);

void yyerror(const char \*s);

%}

%union {

double dval;

}

%token <dval> NUMBER

%type <dval> expression

%left '+' '-'

%left '\*' '/'

%right '^'

%%

input:

| input expression '\n' {

if (floor($2) == $2) {

printf("Result: %.0lf\n", $2);

} else {

printf("Result: %lf\n", $2);

}

}

;

expression:

NUMBER { $$ = $1; }

| expression '+' expression { $$ = $1 + $3; }

| expression '-' expression { $$ = $1 - $3; }

| expression '\*' expression { $$ = $1 \* $3; }

| expression '/' expression {

if ($3 == 0) {

yyerror("Division by zero");

$$ = 0;

} else {

$$ = $1 / $3;

}

}

| expression '^' expression { $$ = pow($1, $3); }

| '(' expression ')' { $$ = $2; }

;

%%

void yyerror(const char \*s) {

fprintf(stderr, "Error: %s\n", s);

}

int main(void) {

printf("Enter expressions:\n");

yyparse();

return 0;

}

**Makefile**

calc: calc.tab.c lex.yy.c

gcc -o calc calc.tab.c lex.yy.c -lm

calc.tab.c calc.tab.h: calc.y

bison -d calc.y

lex.yy.c: calc.l calc.tab.h

flex calc.l

clean:

rm -f calc lex.yy.c calc.tab.c calc.tab.h

**Результат работы программы**

