Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа № 5

По дисциплине ЯПИС

За шестой семестр

Тема: «Компилятор компиляторов»

Вариант 8

Выполнил:

Студент 3 курса

Группы ИИ-16

Журавлёв В.А.

Проверил:

Слинко Е.В.

Брест, 2021

Цель: изучить и опробовать на практике генератор синтаксических анализаторов YACC.

**Задание**

**8.** Входной язык содержит последовательность вызовов процедур, разделенных символом

;(точка с запятой). Вызов процедуры должен состоять из имени процедуры и списка

параметров. В качестве параметров могут выступать идентификаторы и римские цифры

со знаком. (Римскими считать числа записанные большими буквами X, V и I)

**Задание**. Выполнить задание по варианту, реализовав синтаксический анализатор, который для каждой обнаруженной конструкции выдает сообщение на экран.

Код программы:

**lex.l**

%{

#include "bison.tab.h"

#include <stdio.h>

void showError();

%}

%%

"," {return(COMMA);}

";" {return(SEMICOLON);}

"(" {return(L);}

")" {return(R);}

[ \n\t]+ {/\*ignore\*/}

[-]?[XVI]+ {

takeText();

return(NUM);

}

[a-zA-Z][a-zA-Z0-9\_]\* {

takeText();

return(ID);

}

. {showError(); exit(0);}

%%

void takeText(){

yylval.sval = malloc(strlen(yytext));

strncpy(yylval.sval, yytext, strlen(yytext));

yylval.sval[strlen(yytext)] = '\0';

}

void showError(){

printf("Other input");

}

yywrap () {

return (0);

}

**bison.y**

%{

#include <stdio.h>

void yyerror (char const \*);

%}

%union

{

char \*sval;

}

%token L

%token R

%token <sval> ID

%token <sval> NUM

%token <sval> PROCEDURE

%token SEMICOLON

%token COMMA

%start functions

%%

functions: | functions function

;

function: ID L {printf("\nProcedure name: %s\n", $1);}

full\_params

R

SEMICOLON

;

full\_params: /\* empty \*/

| params identifier

;

params: | params param

;

param: identifier COMMA

;

identifier: ID {printf("Identifier : %s\n", $1);}

| NUM {printf("Roman number : %s\n", $1);}

;

%%

void yyerror (char const \*s)

{

printf("Error: %s\n", s);

}

main (void) {

while(1){

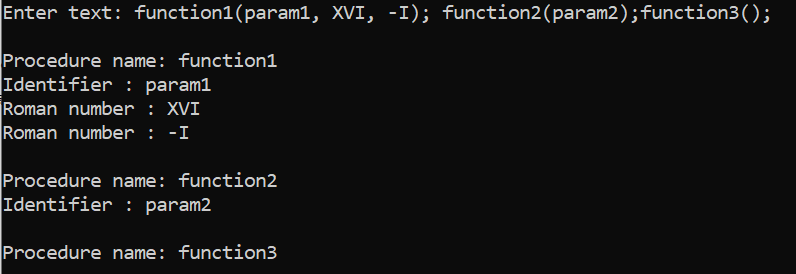
yyparse();

}

return 0;

}

Результат работы программы:



**Вывод:**в ходе данной лабораторной работы изучил и опробовал на практике генератор синтаксических анализаторов YACC.