МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСВЕННЫЙ УНИВЕРСИТЕТ

Кафедра САПР

ОТЧЁТ

о выполнении лабораторной работы №4

по дисциплине «Лингвистическое и программное обеспечение»

Выполнили: ст.гр. 21BA1

Ишенбеков Н.

Жанибекова А.

Женишбек к Э.

Проверил:

Финогеев А.А.

Пенза, 2023г.

**Цель работы:** разработать программу формирования файла мнемокода.

**БНФ реализуемого языка:**

**Вариант 2**

<Программа> ::= <Объявление переменных> <Описание вычислений>

<Описание вычислений> ::= Begin <Список присваиваний> End

<Объявление переменных> ::= Var  <Список переменных> ;

<Список переменных> ::= <Идент> | <Идент> , <Список переменных>

<Список присваиваний>::= <Присваивание> |

                                             <Присваивание> <Список присваиваний>

<Присваивание> ::= <Идент> := <Выражение> ;

<Выражение> ::= <Ун.оп.> <Подвыражение> | <Подвыражение>

<Подвыражение> :: = ( <Выражение> ) | <Операнд> |

                                    < Подвыражение > <Бин.оп.> <Подвыражение>

<Ун.оп.> ::= "-"

<Бин.оп.> ::= "-" | "+" | "\*" | "/"

<Операнд> ::= <Идент> | <Const>

<Идент> ::= <Буква> <Идент> | <Буква>

<Const> ::= <Цифра> <Const>  | <Цифра>

**ADD** - сложение двух верхних элементов стека, результат помещается в вершину стека

**MUL** - умножение двух верхних элементов стека, результат помещается в вершину стека

**SUB** - вычитание элемента в вершине стека из следующего за ним элемента стека,

результат помещается в вершину стека

**DIV** - деление на элемент в вершине стека следующего за ним элемента стека, результат помещается в вершину стека

**LIT** const - засылка константы в стек

**LOAD** **n** - поместить переменную, размещенную по адресу n в вершину стека

**STO** **n** - запись значения из вершины стека по адресу n(присваивание)

**NOT** - инверсия элемента в вершине стека

Листинг программы:

#include <iostream>

#include <fstream>

#include <cctype>

#include <string>

#include <cmath>

#include <vector>

#include <stack>

using namespace std;

//Классы

//1-ключевые слова

//2-идентификаторы

//3-константа

//4-символ присваивания

//5-бинарные операции сложение и вычитание

//6-бинарные операции умножение и деление

//7-унарные операции

//8-символы разделения

//9-скобка открывающая

//10-скобка закрывающая

fstream fi, fo;

vector <string> Gen;

void read\_O() {

char st[10];

int i = 0;

fi >> st;

while (st[0] != ';') {

Gen.push\_back(st);

fi >> st;

}

}

int flag = 0;

vector <string> read\_pr() {

vector <string> str;

int f = 0;

char st[10];

while (f != 1)

{

fi >> st;

if (fi.eof()) { flag = 1; return str; }

str.push\_back(st);

if (st[0] == '=')f = 1;

}

if (fi.eof())flag = 1;

return str;

}

int mem = 0;

bool isIdentifier(string st)

{

for (auto n : st)

{

if (!isalpha(n))

return false;

}

return true;

}

bool isConst(string st)

{

for (auto n : st)

{

if (!isdigit(n))

return false;

}

return true;

}

//ADD - сложение двух верхних элементов стека, результат помещается в вершину стека

//MUL - умножение двух верхних элементов стека, результат помещается в вершину стека

//SUB - вычитание элемента в вершине стека из следующего за ним элемента стека, результат помещается в вершину стека

//DIV - деление на элемент в вершине стека следующего за ним элемента стека, результат помещается в вершину стека

//LIT const - засылка константы в стек

//LOAD n - поместить переменную, размещенную по адресу n в вершину стека

//STO n - запись значения из вершины стека по адресу n(присваивание)

//NOT - инверсия элемента в вершине стека

void main()

{

fi.open("C:\\21VA1\\4 term\\LIPO\\LIPO 3\\postfix.txt", ios\_base::in);

fo.open("mnemocode.txt", ios\_base::out);

if (!fi.is\_open() && !fo.is\_open())

{

cout << "No file!";

exit(2);

}

int flag2 = 0;

int num = 0;

read\_O();

while (flag != 1)

{

vector <string> str;

num = 0;

str = read\_pr();

if (!fi.eof())

{

for (int i = 1; str[i] != "="; i++)

{

if (isIdentifier(str[i]))

{

for (int j = 0; j < Gen.size(); j++)

if (Gen[j] == str[i])num = j + 1;

fo << "LOAD " << num << endl;

}

else if (isConst(str[i]))fo << "LIT " << str[i] << endl;

else if (str[i] == "+") fo << "ADD " << endl;

else if (str[i] == "-")fo << "SUB" << endl;

else if (str[i] == "\*")fo << "MUL" << endl;

else if (str[i] == "/")fo << "DIV" << endl;

else fo << "NOT" << endl;

}

for (int j = 0; j < Gen.size(); j++)

if (str[0] == Gen[j]) { num = j + 1; j = Gen.size(); }

fo << "STO " << num << endl;

str.clear();

}

else flag = 1;

}

fi.close();

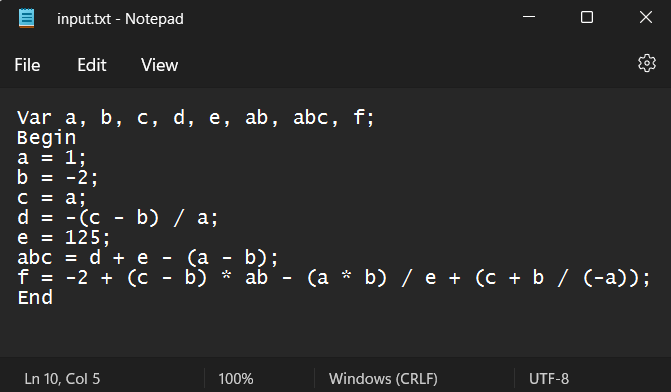
fo.close();

system("pause");

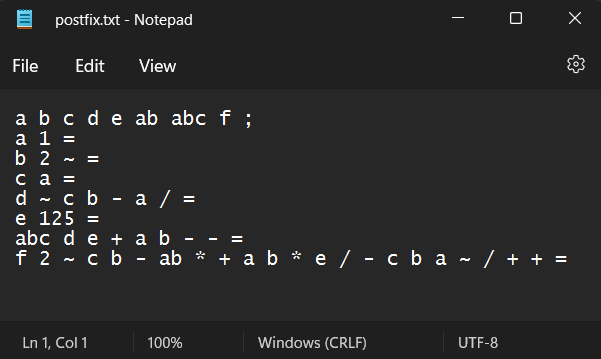
return;

}

**Результат работы программы:**

Исходный файл:  


Постфиксная запись:



Полученный файл:

