Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №4**

З дисципліни «СП»

Виконав: Перевірив:

Студент групи ІО-21 доц. Пустоваров В.І.

Коноз А.О.

Дата здачі\_\_\_\_\_\_\_\_\_\_\_\_\_

Захищено з балом\_\_\_\_\_

Київ 2014

**Завдання**

|  |  |  |
| --- | --- | --- |
| 9 | b=c?d:2\*a[n]; | С |

**Лістинг коду**

// spLb5.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include "..\spLb3\token.h"

#include "..\spLb3\visgrp.h"

#include "..\spLb4\tables.h"

#include "..\spLb4\lexan.h"

#include "syntaxP.h"

#include "..\spLb4\langio.h"

#include <stdio.h>

#include <string.h>

extern struct recrdKWD tablKWD[];

extern struct lxNode nodes[MAX\_NODES]; // масив приймач вузлів дерева

extern enum ltrType ltClsC[256];

extern enum ltrType ltClsP[256];

extern enum tokType dlCdsC[256];

extern enum tokType dlCdsP[256];

extern enum ltrType ltClsC[256];

extern enum ltrType ltClsP[256];

enum ltrType \*ltCls=ltClsC;

enum tokType \*dlCds=dlCdsC;

char file\_name[20];

int main(int argc, char\* argv[])

{int nn=-1, nr=0, nc=1; //np,

if (argc>1)

{strcpy(file\_name,argv[1]);

printf("Processing file -- %s\n",file\_name);}

else

{printf("Please enter file Name: ");

scanf("%s",file\_name);

strcat(file\_name,".h");

}

opFls(file\_name);

LxAnInit('C');

// srtBin(tablKWD, 67);

do{//np=nn;

nn=LxAnlzr();

}while(nodes[nn].ndOp!=\_EOF);

prLaTxt(nodes,nn);

printf("\n");

SxAnInit('C');

nr=0; nc=1; nodes[0].prnNd=-1;

do nr=nxtProd(nodes,nr,nc);

while(++nc<nn);

// конверсія до семантичної обробки

prLxTxt(nodes+nr);

// nr=prCmpr(nodes,nn,nr);// компресія для скорочення графа

// prLxTxt(nodes+nr);

printf("\n");

return 0;

}

#include "stdafx.h"

#include "tables.h"

#include <string.h>

// порівняння рядків

int cmpStr(unsigned char\* s1, unsigned char\* s2)

{unsigned n=0;

while(s1[n]==s2[n]&&s1[n]!=0)n++;

return s1[n]-s2[n];

}

// порівняння за відношенням порядку

int cmpKys(char \*k0, char \*kArg)

{int i=cmpStr((unsigned char\*)k0,

(unsigned char\*)kArg);

//if(i)

return i;

}

//------------------------------------------------------

// вибірка за двійковим пошуком

struct recrdKWD\*selBin(char\* kArg, struct recrdKWD\*tb, int ln)

{int i, nD=-1, nU=ln, n=(nD+nU)>>1;

while(i=cmpKys(tb[n].key,kArg))

{if(i>0)nU=n;else nD=n;

n=(nD+nU)>>1;

if(n==nD)return NULL;

}

return &tb[n];

}

// сортування для двійкового пошуку

struct recrdKWD\*srtBin(struct recrdKWD\*tb, int ln)

{int n=0, n1;

struct recrdKWD el;

for(;n<ln;n++)for(n1=n+1;n1<ln;n1++)

if(cmpKys(tb[n].key,tb[n1].key)>0)

{el=tb[n];tb[n]=tb[n1];tb[n1]=el;}

return tb;

}

#include "stdafx.h"

#include "syntaxP.h"

#include "..\spLb3\token.h"

#include <stdio.h>

struct lxNode nodes[MAX\_NODES]= // масив приймач вузлів дерева

{{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},

{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},

{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},

{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},

{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},

{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},

{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},

{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},

{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},

{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},{\_nil,NULL,NULL,0,0,0,0,0,NULL,0},

};

#include "stdafx.h"

#include "automat.h"

#include "tables.h"

#include "..\spLb3\parsC.cpp"

//char

enum tokType dlCdsC[256]=

{\_EOF,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil, //16

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil, //32

\_nil,\_xmrk,\_nil,\_nil/\*#\*/, \_nil/\*$\*/,\_mod,\_andB,\_nil,

\_brkt,\_bckt,\_mul,\_add, \_comma,\_sub,\_fldDt,\_div, //48

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_cln,\_EOS, \_lt,\_ass,\_gt,\_qmrk,// 64

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,//80

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_ixbr, \_nil,\_scbr,\_xorB,\_nil,//96

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,//112

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_opbr, \_orB,\_ocbr,\_invB,\_nil,//128

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,//144

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,//160

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,//176

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,//192

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,//208

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,//224

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,//240

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,

\_nil,\_nil,\_nil,\_nil, \_nil,\_nil,\_nil,\_nil,//256

};

enum ltrType ltClsC[256] =

{dlmaux,nc,nc,nc,nc,nc,nc,nc, nc,dlmaux,dlmeormr,nc,nc,dlmeormr,nc,nc, //16

nc,nc,nc,nc,nc,nc,nc,nc, nc,nc,nc,nc,nc,nc,nc,nc, //32

dlmaux,dlmgrop,ltrstrlm,dlmunop,

ltrnmelm,dlmgrop,dlmgrop,ltrstrlm,

dlobrct,dlcbrct,dlmgrop,ltrsign,

dlmunop/\*dlmbrlst\*/,ltrsign,dldot,dlmgrop, //48

dgt,dgt,dgt,dgt,dgt,dgt,dgt,dgt,

dgt,dgt,dlmgrop,dlmunop/\*dlmbrlst\*/,dlmgrop,dlmgrop,dlmgrop,dlmunop,// 64

dlmunop,ltrhxdgt,ltrhxdgt,ltrhxdgt, ltrhxdgt,ltrexplt,ltrhxdgt,ltrnmelm,

ltrnmelm,ltrnmelm,ltrnmelm,ltrnmelm, ltrnmelm,ltrnmelm,ltrnmelm,ltrnmelm,//80

ltrnmelm,ltrnmelm,ltrnmelm,ltrnmelm, ltrnmelm,ltrnmelm,ltrnmelm,ltrnmelm,

ltrtpcns/\*ltrnmelm\*/,ltrnmelm,ltrnmelm,dlobrct, ltrtrnfm,dlcbrct,dlmgrop,ltrnmelm,//96

dlmunop,ltrhxdgt,ltrhxdgt,ltrhxdgt, ltrhxdgt,ltrexplt,ltrhxdgt,ltrnmelm,

ltrnmelm,ltrnmelm,ltrnmelm,ltrnmelm, ltrnmelm,ltrnmelm,ltrnmelm,ltrnmelm,//112

ltrnmelm,ltrnmelm,ltrnmelm,ltrnmelm, ltrnmelm,ltrnmelm,ltrnmelm,ltrnmelm,

ltrtpcns/\*ltrnmelm\*/,ltrnmelm,ltrnmelm,dlobrct, dlmgrop,dlcbrct,dlmunop/\*dlmgrop\*/,nc,//128

nc,nc,nc,nc,nc,nc,nc,nc, nc,nc,nc,nc,nc,nc,nc,nc, //144

nc,nc,nc,nc,nc,nc,nc,nc, nc,nc,nc,nc,nc,nc,nc,nc, //160

nc,nc,nc,nc,nc,nc,nc,nc, nc,nc,nc,nc,nc,nc,nc,nc, //176

nc,nc,nc,nc,nc,nc,nc,nc, nc,nc,nc,nc,nc,nc,nc,nc, //192

nc,nc,nc,nc,nc,nc,nc,nc, nc,nc,nc,nc,nc,nc,nc,nc, //208

nc,nc,nc,nc,nc,nc,nc,nc, nc,nc,nc,nc,nc,nc,nc,nc, //224

nc,nc,nc,nc,nc,nc,nc,nc, nc,nc,nc,nc,nc,nc,nc,nc, //240

nc,nc,nc,nc,nc,nc,nc,nc, nc,nc,nc,nc,nc,nc,nc,nc //256

//...

};

enum autStat nxtStsC[Eo+1][ltrcode+1] = // for C/C++

{{Eu,Eu,Eu,Eu,Eu, Eu,Eu,Eu,S0,S2, S0,S2,S0,S0,S0, S0,S0,S0}, //0 для Eu

{S1c,S1n,S1n,S1n,S1n, S1s,Soc,Eu,S0,S2, S0,S2,S0,S0,S0, S0,S0,S0}, //1 для S0 -> S0c

{S1c,Ec,Ec,Ec,Ec, Ec,Soc,Eu,S2c,S2, S0,S2,S0,S0,S0, S0,S0,S0}, //2 для S1g

{S1c,S1e,Ec,Ec,Ec, Ec,Soc,Eu,S2c,S2, S0,S2,S0,S0,S0, S0,S0,S0}, //3 для S1c

{S2c,S1e,Ec,Ec,Ec, Ec,Soc,Eu,Ec,S2, S0,S2,S0,S0,S0, S0,S0,S0}, //4 для S2c

{S1p,Ec,Ec,Ec,Ec, Ec,Soc,Eu,S0,S1q, S0,S2,S0,S0,S0, S0,S0,S0}, //5 для S1e

{S1p,Ec,Ec,Ec,Ec, Ec,Soc,Eu,Ec,Ec, S0,S2,S0,S0,S0, S0,S0,S0}, //6 для S1q

{S1p,Ec,Ec,Ec,Ec, Ec,Soc,Eu,S0,S2, S0,S2,S0,S0,S0, S0,S0,S0}, //7 для S1p

{S1n,S1n,S1n,S1n,S1n, En,S1n,En,S0,S2, S0,S2,S0,S0,S0, S0,S0,S0}, //8 для S1n

{S1s,S1s,S1s,S1s,S1s, S2s,S1t,S1s,S1s,S1s, S1s,S1s,S1s,S1s,S1s, S1s,S1s,S1s},//9 для S1s

{S1t,Ec,S1s,S1s,S1s, S2s,S1s,Ec,S0,S2, S0,S2,S0,S0,S0, S0,S0,S0}, //10 для S1t

{S1s,Ec,S1s,S1s,S1s, S1s,S1t,Ec,S0,S2, S0,S2,S0,S0,S0, S0,S0,S0}, //11 для S2s

{S1c,S1n,S1n,S1n,S1n, S1s,Soc,Eu,S0,S3, S0,S3,S0,S0,S0, S0,S0,S0}, //12 для S2 - Початковий елемент групового роздільника

{S1c,S1n,S1n,S1n,S1n, S1s,Soc,Eu,S0,S3, S0,S3,S0,S0,S0, S0,S0,S0}, //13 для S3 - Наступний елемент групового роздільника

{S0p,Ec,Ec,S0p,Ec,Ec,Ec,Ec, S0,S0,S0}, //14 для S3c

{S3c,Ec,Ec,S3c,Ec,Ec,Ec,Ec, S0,S0,S0}, //15 для S0p

{Soc,S3c,S3c,Ec,Ec,Ec,Ec,Eu, S2c,S0,S0}, //16 для Soc- Вісімковий код

{Scr,Scr,Scr,Scr,Scr, Scr,Scr,Scr,Scr,Scr, Scr,Scr,Scr,Scr,Scr, Scr,S0,Scr},//17 для Scr- Коментар-рядок

{Scl,Scl,Scl,Scl,Scl, Scl,Scl,Scl,Scl,Scl, Scl,Scl,Scl,Scl,Scl, Scl,Scl,Scl}, // для Scl

{Ec,Ec,Ec,Ec,Ec,Ec,Ec,Ec, S0,S0,S0}, // для Ec

{Ep,Ep,Ep,Ep,Ep,Ep,Ec,Ec, S0,S0,S0}, // для Ep

{Eq,Eq,Eq,Eq,Eq,Eq,Eq,Ec, S0,S0,S0}, // для Eq

{En,En,Eq,Eq,Eq,Eq,Eq,Ec, S0,S0,S0}, // для En

{Eo,Eo,Eo,Eq,Eq,Eq,Eq,Ec, S0,S0,S0} // для Eo...

};

struct recrdKWD tablKWDC[67]=

{{oprtrC[\_ne],\_ne,1}, {oprtrC[\_asMod],\_asMod,1},

{oprtrC[\_and],\_and,1}, {oprtrC[\_asAnd],\_asAnd,1},

{oprtrC[\_asMul],\_asMul,1},{oprtrC[\_inr],\_inr,1},

{oprtrC[\_asAdd],\_asAdd,1},{oprtrC[\_dcr],\_dcr,1},

{oprtrC[\_asSub],\_asSub,1},{oprtrC[\_fldPt],\_fldPt,1},

{oprtrC[\_rem],\_rem,1},{oprtrC[\_remL],\_remL,1},// 10

{oprtrC[\_asDiv],\_asDiv,1},{oprtrC[\_shLft],\_shLft,1},

{oprtrC[\_asShl],\_asShl,1},{oprtrC[\_le],\_le,1},

{oprtrC[\_eq],\_eq,1},{oprtrC[\_ge],\_ge,1}, // 16

{oprtrC[\_shRgt],\_shRgt,1},{oprtrC[\_asShr],\_asShr,1},

{oprtrC[\_asXor],\_asXor,1},{oprtrC[\_auto],\_auto,1},// 20 подвыйна двокрапка

{oprtrC[\_break],\_break,1},

{oprtrC[\_switch],\_switch}, {oprtrC[\_catch],\_catch,6},

{oprtrC[\_char],\_char,1}, {oprtrC[\_class],\_class,2},

{oprtrC[\_const],\_const,1}, {oprtrC[\_continue],\_continue,1},

{oprtrC[\_default],\_default,1},{oprtrC[\_delete],\_delete,2},

{oprtrC[\_repeat],\_repeat,1},{oprtrC[\_double],\_double,1},

{oprtrC[\_else],\_else,1},

{oprtrC[\_enum],\_enum}, {oprtrC[\_extern],\_extern},

{oprtrC[\_float],\_float}, {oprtrC[\_for],\_for},

{oprtrC[\_friend],\_friend}, {oprtrC[\_goto],\_goto},

{oprtrC[\_if],\_if}, {oprtrC[\_int],\_int},

{oprtrC[\_sint64],\_sint64}, {oprtrC[\_long],\_long},

{oprtrC[\_new],\_new}, {oprtrC[\_private],\_private},

{oprtrC[\_protected],\_protected},{oprtrC[\_public],\_public},

{oprtrC[\_register],\_register},{oprtrC[\_return],\_return},

{oprtrC[\_short],\_short}, {oprtrC[\_signed],\_signed},

{oprtrC[\_sizeof],\_sizeof}, {oprtrC[\_static],\_static},

{oprtrC[\_struct],\_struct}, {oprtrC[\_case],\_case,1},

{oprtrC[\_throw],\_throw}, {oprtrC[\_tryC],\_tryC},

{oprtrC[\_typedef],\_typedef},{oprtrC[\_union],\_union},

{oprtrC[\_unsigned],\_unsigned},{oprtrC[\_virtual],\_virtual},

{oprtrC[\_void],\_void}, {oprtrC[\_volatile],\_volatile},

{oprtrC[\_whileP],\_whileP},

{oprtrC[\_asOr],\_asOr,1},{oprtrC[\_or],\_or,1},

};

#include "stdafx.h"

#include "automat.h"

enum autStat nxtSts[Se+1][sgE+1] =

{

{S0, S0, S0, S7, S0, S0, S0, S0, S0, S0}, // для S0

{S0, S0, S2, S3, S4, S5, S6, Se, S3, S3}, // S1

{Se, Se, Se, Se, Se, Se, Se, Se, Se, Se}, // S2

{Se, Se, Se, Se, Se, Se, Se, Se, Se, Se}, // S3

{Se, Se, Se, Se, Se, Se, Se, Se, Se, Se},

{Se, Se, Se, Se, Se, Se, Se, Se, Se, Se},

{S1, S1, S2, S3, S4, S5, S6, Se, S3, S3},

{S5, S5, S5, S5, S5, S5, S5, S5, S5, S5},

{S5, S5, S5, S5, S5, S5, S5, S5, S5, S5}, // Se

};

enum autStat nxtStat(enum autSgn sgn)

{

static enum autStat s = S0; // поточний стан лексеми

return s = nxtSts[s][sgn]; // новий стан лексеми

}

#define begOprtr 0x50+46+78 // зміщення початку виконавчих операторів

#define \_lVlu 0x1000 // lValue

enum tokType

{\_nil, \_nam, //0 зовнішнє подання

\_srcn, \_cnst, //2 вхідне і внутрішнє кодування константи

\_if,\_then,\_else,\_elseif, //4 if then else elseif

\_case, \_switch, \_default, \_endcase,//8 case switch defualt endcase

\_break, \_return, \_whileP, \_whileN, //12 break return while do

\_continue, \_repeat, \_untilN, \_endloop, //16 continue repeat until

\_for, \_to, \_downto, \_step,// for to downto step

\_untilP, \_loop, \_with, \_endif,

\_void,\_extern,\_var,\_const,\_enum,\_struct/\*\_record\*/,\_union,\_register,//

\_unsigned,\_signed,\_char,\_short,\_int,\_long,\_sint64,\_uint64,//

\_float,\_double, \_label,\_auto,\_static,\_volatile,\_typedef,\_sizeof,//

\_real,\_array,\_set,\_file,\_object, \_string,\_goto,

\_program,\_function,\_procedure /\*task V\*/,

\_macromodule,\_primitive,\_specify,\_table, //Verilog

\_generate,\_config,\_liblist,\_library, //Verilog

\_incdir,\_include,\_design,\_defaultS,\_instance,\_cell,\_use, //Verilog

\_automatic,\_endmodule,\_endfunction,\_endtask, //Verilog

\_endprimitive,\_endspecify,\_endtable,\_endgenerate,\_endconfig, //Verilog

\_endcaseV,\_casex,\_casez,\_wait,\_forever,\_disable,\_ifnone, //Verilog

\_pulsestyle\_onevent,\_pulsestyle\_ondetect,\_showcanceled,\_noshowcanceled, //Verilog

\_vectored,\_scalared,\_small,\_medium,\_large, //Verilog

\_genvar,\_parameter,\_localparam,\_defparam,\_specparam,\_PATHPULSE$, //Verilog

\_inlineF,\_forward,\_interrupt,\_exportF,\_extrn,\_asmb,

\_input,\_output,\_inout, //Verilog|SQL+3

\_objectP,\_constructor,\_desctructor,\_property,\_resP,\_abstract, //P++9

\_class,\_public,\_private,\_protected,\_virtual,\_friend, //C++16

\_new,\_delete,\_tryC,\_catch,\_throw/\*raise\*/, //C++20

\_initial,\_always,\_assign,\_deassign,\_force,\_release, //Verilog+26

\_reg,\_time,\_realtime,\_event,\_buf,\_not, //Verilog+32

\_andG,\_orG,\_xorG,\_nandG,\_norG,\_xnorG, //Verilog+38

\_tran,\_tranif0,\_tranif1,\_rtran,\_rtranif0,\_rtranif1, //Verilog+44

\_tri,\_trior,\_triand,\_trireg,\_tri0,\_tri1,//Verilog+50

\_wire,\_wand,\_wor,\_wres, //Verilog+54

\_supply0,\_supply1,\_highz0,\_highz1, //Verilog+58

\_strong0,\_strong1,\_pull0,\_pull1,\_weak0,\_weak1, //Verilog+64

\_pulldown,\_pullup,\_bufif0,\_bufif1,\_notif0,\_notif1, //Verilog+70

\_cmos,\_rcmos,\_nmos,\_pmos,\_rnmos,\_rpmos, //Verilog+76

\_fork, \_join, // відкриті і закриті дужки паралельних операторів 2

\_opbr, \_ocbr, // відкриті і закриті дужки операторів 2

\_ctbr, \_fcbr, // відкриті і закриті дужки конкатенацій 3

\_ixbr, \_scbr, // відкриті і закриті дужки індексу 4

\_brkt, \_bckt, // відкриті і закриті дужки порядку і функцій 5

\_tdbr, \_tcbr, // відкриті і закриті дужки даних 6

\_eosP, eosS, // паралельні та послідовні

\_EOS=begOprtr, \_comma, \_cln, \_qmrk,// ; , : ?

\_asOr, \_asAnd, \_asXor, \_asAdd, //|= =& =^ =+

\_asSub, \_asMul, \_asDiv, \_asMod, // -= \*= /= %=

\_asShr,\_asShl, \_ass, \_dcr, \_inr, // <<= >>= = -- ++

\_dcrN,\_inrN,\_mcrs,\_dbcln,\_eoCm,\_EOF, //-- ++ // # :: \*/

\_lt,\_le, \_eq, \_ne, \_ge,\_gt, // < <= == != >= >

\_add, \_sub, \_mul, \_div, \_fldDt, \_fldPt,// + - \* / . ->

\_pwr, \_shLfa, \_shRga, \_eqB, \_neB, // \*\* <<< >>> === !==

\_addU,\_subU,\_refU, \_ptrU, // + - \* & унарні

\_lmts,\_eqar,\_astar,\_trasand, // PV+4 .. => \*> &&&

\_orR,\_andR,\_xorR,\_norR,\_nandR,\_nxorR,\_xornR, //V+11 & | ^ ~| ~& ~^

\_delay,\_eventV,\_events, //V+14 # @ @\*

\_norB,\_nandB,\_nxorB,\_xornB,\_addr, //~| ~& ~^ ^~ \_ptr,

\_rem,\_remL, //

\_mod, \_orB, \_andB, \_xorB, // %(mod) |(or) &(and) ^(xor)

\_shLft,\_shRgt, \_or, \_and, //<<(shl) >>(shr) ||(or) &&(and)

\_xmrk,\_invB,\_divI,\_in, //\_not, \_notB, /(div)

\_posedge,\_negedge,\_orE, //Verilog+3

\_frkz, // відкриті і закриті дужки паралельних операторів 2

\_opbz, // відкриті і закриті дужки операторів 2

\_ctbz, // відкриті і закриті дужки конкатенацій 3

\_ixbz, // відкриті і закриті дужки індексу 4

\_brkz, // відкриті і закриті дужки порядку і функцій 5

\_tdbz // відкриті і закриті дужки даних 6

// \_pnil

};

struct lxNode//вузол дерева, САГ або УСГ

{enum tokType ndOp; //код типу лексеми

struct lxNode\* prvNd;// зв'язок з попередником

struct lxNode\* pstNd;// зв'язок з наступником

int dataType; // код типу даних, які повертаються

unsigned resLength; //довжина результату

int x, y, f;//координати розміщення у вхідному файлі

int prnNd;//struct lxNode\* prnNd;//зв'язок з батьківським вузлом

unsigned stkLength;//довжина стека обробки семантики або номер модуля

};

// порівняння за відношенням порядку

int cmpKys(struct lxNode\*, struct lxNode\*);

**Результат виконання програми:**

Microsoft Windows [Version 6.1.7601]

Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>"E:\Education\5\SP\spLb1 - Copy\spLb1\spLb5\Debug\spLb5.exe"

Please enter file Name: E:\myfunction

b=c?d:2\*a[ n];

b=c?d:2\*(a[n]);

C:\Users\User>

**Висновок:**

Під час виконання лабораторної роботи були отримані навички налаштування таблиць висхідних синтаксичних аналізаторів та програм побудови графів та дерев розбору на етапі синтаксичного аналізу з запам’ятовуванням покажчиків на вхідні образи та внутрішні коди. Також були вивчені програми формування повідомлень про помилки та побудова графа підлеглості операцій та операторів в процесі висхідного синтаксичного аналізу.