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 №3**

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**

// spLb4.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

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

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

#include "tables.h"

#include "lexan.h"

#include "langio.h"

#include <stdio.h>

#include <string.h>

extern struct recrdKWD \*tablKWD;

struct lxNode nodes[200]= // 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},

};

char file\_name[20];

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;

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

{int nn=-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");

return 0;

}

#include "stdafx.h"

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

#include "tables.h"

#include "index.h"

#include <stdlib.h>

// порівняння рядків

// порівняння терміналів за відношенням порядку

int cmpTrm(struct lxNode\*k0,struct lxNode\*kArg)//cmpKys

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

(unsigned char\*)kArg->prvNd);

if(i)return i;

return k0->stkLength - kArg->stkLength; // порівняння номерів модулів

}

unsigned nNdxNds=0;

struct indStrUS ndxNds[50]={{NULL,NULL,NULL,0}},

\*pRtNdx=ndxNds, nilNds={NULL,NULL,NULL,0};

// вибірка через пошук за двійковим деревом

struct indStrUS\*selBTr(struct lxNode\*kArg,struct indStrUS\*rtTb)

{int df;

while(df=cmpTrm(kArg,rtTb->pKyStr))

if(df>0){if(rtTb->pRtPtr)rtTb=rtTb->pRtPtr;

else break;}

else{if(rtTb->pLtPtr)rtTb=rtTb->pLtPtr;

else break;}

rtTb->dif=df;

return rtTb;

}

// включення через пошук за двійковим деревом

struct indStrUS\*insBTr(struct lxNode\*pElm,struct indStrUS\*rtTb)

{struct indStrUS\*pInsNod;//,\*pNod;

if(rtTb->pKyStr==NULL)

{rtTb->pKyStr=pElm;

return rtTb;

}

// if(rtTb->pKyStr->ndOp==\_nil)rtTb->pKyStr=pElm;

else{pInsNod=selBTr(pElm,rtTb);

if(pInsNod->dif)

{ndxNds[++nNdxNds]=nilNds;

if(pInsNod->dif<0)pInsNod=pInsNod->pLtPtr=ndxNds+nNdxNds;

else pInsNod=pInsNod->pRtPtr=ndxNds+nNdxNds;

ndxNds[nNdxNds].pKyStr=pElm;

}

}

return pInsNod;

}

#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;

}

enum ltrType

{dgt, //0 десяткова цифра

ltrexplt,//1 літера-ознака експоненти

ltrhxdgt,//2 літера-шістнадцяткова цифра

ltrtpcns,//3 літера-визначник типу константи

ltrnmelm,//4 літери, які припустимі тільки в іменах

ltrstrlm,//5 літери для обмеження рядків і констант

ltrtrnfm,//6 літери початку перекодування літер рядків

nc, //7 некласифіковані літери

dldot, //8 точка як роздільник та літера констант

ltrsign, //9 знак числа або порядку

dlmunop, //10 поодинокі роздільники операцій

dlmgrop, //11 елемент групового роздільника

dlmbrlst,//12 роздільники елементів списків

dlobrct, //13 відкриті дужки

dlcbrct, //14 закриті дужки

dlmeorml,//15 кінець обмеженого коментаря

dlmeormr,//16 кінець коментаря-рядку

dlmaux, //17 допоміжні роздільники типа пропусків

ltrcode=18// ознака можливості вісімкового кодування

};

enum autStat

{Eu, //0 Eu - Некласифікований об'єкт

S0, //1 S0 - Роздільник

S1g, //2 S1g - Знак числової константи

S1c, //3 S1c - Ціле число

S2c, //4 S2c - Число з точкою

S1e, //5 S1e - Літера "e" або "E"

S1q, //6 S1q - Знак "-" або "+"

S1p, //7 S1p - Десяткові цифри порядку

S1n, //8 S1n - Елементи імені

S1s, //9 S1s - Літери рядка або символьної константи

S1t, //10 S1t - Елементи констант, які перетворюються

S2s, //11 S2s - Ознака закінчення константи

S2, //12 S2 - Початковий елемент групового роздільника

S3, //13 S3 - Наступний елемент групового роздільника

S3c, //14?S3c - Ціле число з недесятковою основою

S0p, //15?S0p - Ознака типу константи

Soc, //16 Soc- Вісімковий код

Scr, //17 Scr- Коментар-рядок

Scl, //18 Scl- Обмежений коментар

Ec, //19 Ec - Неправильна константа

Ep, //20 Ep - Неправильна константа з точкою

Eq, // Eq - Неправильна константа з порядком

En, // En - Неправильне ім'я

Eo // Eo - Неприпустиме сполучення операцій

};

**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\spLb4\Debug\spLb4.exe"

Please enter file Name: E:\myfunction

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

C:\Users\User>

**Висновок:**

Під час виконання лабораторної роботи була вивчена схема табличного подання автоматної граматики лексичного аналізу. Також були засвоєні навички використання об’єктів стану графів автоматів для формування лексем у форматі внутрішнього подання вузлів графу розбору.