Министерство образования и науки Российской Федерации

Федеральное Государственное Бюджетное Образовательное Учреждение

Высшего Профессионального Образования

Тамбовский Государственный Технический Университет

Кафедра

Отчёт по лабораторной работе №2

по дисциплине «Лингвистические средства вычислительных систем»

Вариант 3

Выполнил студент гр.

.

Проверила:.

Тамбов 20

Постановка задачи:

Произвести синтаксический анализ методом рекурсивного спуска.

D=B\*B-4\*A\*C

if(D.EQ.O)T=B/(2\*A)

TYPE \*; T

Кодировочная таблица:

|  |  |  |
| --- | --- | --- |
| Лексема | Код | |
| If | 1 | |
| EQ | 2 | |
| TYPE | 3 | |
| . | 4 | |
| \* | 5 | |
| ( | 6 | |
| ) | 7 | |
| = | 8 | |
| - | 9 | |
| / | 10 | |
| ; | | 11 |
| Идентификатор | | 12 |
| D | | 12 1 |
| B | | 12 2 |
| A | | 12 3 |
| C | | 12 4 |
| T | | 12 5 |
| O | | 12 6 |
| Константа | | 13 |
| 4 | | 13 4 |
| 2 | | 13 2 |

Грамматика языка:

<список операторов> → <оператор> {конец строки, <оператор>}

<оператор> → <присваивание> / <условный оператор> / <вывод>

<присваивание> → идентификатор = <арифметическое выражение>

<арифметическое выражение> → <слагаемое> { - , <слагаемое>}

<слагаемое> → <множитель> {\*, /, <множитель>}

<множитель> → идентификатор / константа / (<слагаемое>)

<условный оператор> → if (<условие>) <присваивание>

<условие > → идентификатор. EQ. идентификатор

<вывод> → TYPE \*; идентификатор

Результат:

BB\*4AC\*\*-D=B2A\*()/T=if()TYPE\*;T

Листинг программы:

#include <stdio.h>

int mnoj(int j, int n, char \* str)

{

if(str[j+2]!='\*' && str[j-2]!='\*')

printf("%c%c",str[j-1], str[j+1]);

else if(str[j+2]=='\*') {printf("%c%c%c", str[j-1], str[j+1], str[j+3]); j=j+3;}

return 0;

}

int clag(int j, int n, char \* str)

{

int k=n;

while (str[k]!='\n')

{

k++;

}

for(int i=n; i<k; i++)

{

if(str[i]=='\*')

if(str[i]=='\*') {mnoj(i,n,str); printf("\*");}

if(str[i]=='/')

{ printf("%c",str[i-1]);

if(str[i+1]=='(')

{n=i+1; i=i+3; n=i; clag(i,n,str); printf("()");}

printf("/");

}

}

return 0;

}

int ar(int j, char \* str)

{

int n=j,fl=0,i;

for (i=j; i<'\n'; i++)

{

if(str[i]=='-')

{

clag(i,n,str);

printf("-");

fl=1;

}

}

if(fl==0) {i=n; while(str[i]!='\n')

{i++;}

clag(i,n,str); }

return 0;

}

int pris(int j, char \*str)

{ int k;

k=j; ar(j, str); printf("%c",str[k-1]);

return 0;

}

int YO(int j, char \*str)

{

int i=j+1;

while(str[i]!=')')

{

printf("%c", str[i]);

i++;

}

pris(i+2,str);

return 0;

}

int out(int j, int n, char \* str)

{

while (j<n)

{

printf("%c", str[j]);

j++;

}

return 0;

}

int OP(int n, char \* str)

{

int k;

for (int i=0; i<n; i++)

{

if(str[i]=='=')

{

pris(i, str);

printf("=");

}

else if(str[i]=='i')

{

YO(i+2,str);

printf("if()");

while(str[i]!='\n')

{

i++;

}

}

else if(str[i]=='T' && str[i+1]=='Y')

{

out(i,n,str);

}

}

return 0;

}

int main()

{

FILE \*f=fopen("in.txt", "r");

fseek(f,1,SEEK\_END);

int n = ftell (f);

n=n-1;

rewind(f);

char \*str= new char [n];

for (int i=0; i<n; i++)

{

fscanf(f, "%c", &str[i]);

if(str[i]=='\n' || str[i]==' ') {

n--;}

}

OP(n,str);

return 0;

}