Лабораторна робота №2, ІП-31, Кахерський О.І.

**Формирование постфиксной записи**

**Цель:** разобраться с преимуществами постфиксной записи с точки зрения ее применения в компиляторах, сформировать умения и навыки формирования постфиксной записи, преобразования инфиксной записи выражения в постфиксную.

Вариант задания

1. Подмножество языка PASCAL включает:

* данные типа **int, float, struct** из элементов указанных типов;
* инструкции описания переменных;
* операторы присваивания в любой последовательности;
* операции **+, – , \*, = =, != , <, >** .

Листинг программы

//пріорітети виконання операторів

var priorities = [ [[";"],-4],

[["%"],-5],

[["(",")"],-2],

[["=",":",","],-3],

[["?","!"],1],

[[">","<"],2],

[["+","-"],3],

[["\*","/"],4]];

//стек операторів

var functionStack = [];

//постфіксна форма

var resultStatement = "";

// функція переводить всі арифметичні вирази та базові оператори в постфіксну форму.

function reformatArifmeticOperator(statement){

var splittedStatement = statement.split('');

splittedStatement.forEach(function(symbol){

processNewSymbol(symbol);

});

functionStack.reverse().forEach(function(operator){

resultStatement += operator;

});

resultStatement+="\n"

}

// функція переводить оператор record мови Паскаль в постфіксну форму

function reformatStructOperator(statement){

reformatArifmeticOperator(statement.

replace("type","%").

replace(/=record|= record/g,"(").

replace("end;",')').

replace(/\n|\t/g,''));

resultStatement = resultStatement.replace("%","struct")

}

function reformatIfOperator(statement){

//очищення вхідного рядка від, заміна багатосимвольних операторів на односимвольні

var splittedStatement = statement.

replace(/begin/g,"(").

replace(/end;/g,")").

replace(/if|else|then|/g,'').

replace(/\s/g,"").

replace(/==/g,"?").replace(/!=/,"!").replace(/:=/g,'=').

split('');

//системні мітки. Використовуються при обчисленні гілок оператору if (робить можливим обчислення одної гілки).

//@BZ - перехід на мітку @, якщо значення в if - false. #BR - перехід на мітку # при true

var systemLinks = ["@BZ","#BR@","#"];

var sysCounter = 0;

splittedStatement.forEach(function(symbol){

processNewSymbol(symbol);

if(symbol == ")")

resultStatement += systemLinks[sysCounter++];

});

functionStack.reverse().forEach(function(operator){

resultStatement += operator;

});

resultStatement = resultStatement.replace(/\=/g,":=").replace(/\?/g,"==").replace(/\!/g,"!=")

resultStatement+="\n"

}

//визначає пріорітет оператору

function getPriority(symbol){

var result = -1;

priorities.forEach(function(item){

if(item[0].indexOf(symbol) !== -1){

result = item[1];

return;

}

});

return result;

}

//додає новий оператор до стеку, оператор видаляє зі стеку всі оператори, що мають пріорітет, нижчий або рівний даному

function pushNewOperator(symbol,priority){

var closeSymbol = symbol == ")";

for(i = functionStack.length - 1; i >= 0; i--){

if(getPriority(functionStack[i]) >= priority && symbol != '('){

if(functionStack[i] != '(')

resultStatement += functionStack.pop();

else {

functionStack.pop();

if(closeSymbol)

break;

}

}

else break;

}

if(!closeSymbol)

functionStack.push(symbol);

}

//функція обробляє новий символ (з'ясовує, чи є цей символ оператором чи це операнд)

function processNewSymbol(symbol){

var priority = getPriority(symbol);

if(priority == -1)

resultStatement += symbol;

else pushNewOperator(symbol,priority);

}

//тести переводу в постфіксну форму арифметичних виразів, операторів присвоювання, умовного оператору та record

reformatArifmeticOperator("**a+b\*c\*d+(e-f)\*(g\*h+i)**");

console.log(resultStatement);

resultStatement =''

reformatArifmeticOperator("**c,b,a: Integer**;");

console.log(resultStatement);

resultStatement=""

reformatArifmeticOperator**("(a+b)\*c-(d-e)\*(f+g)**");

console.log(resultStatement);

resultStatement=''

reformatIfOperator("**if(a==b) then begin a:=a+b; end; else begin a:=a-b; end;**");

console.log(resultStatement);

resultStatement=""

reformatStructOperator("**type rec = record\na: Integer;\nb: float\nend;**");

console.log(resultStatement);

Результат работы программы

abc\*d\*+ef-gh\*i+\*+

c+\*b,a, Integer:;

ab+c\*de-fg+\*-;

ab==@BZaab+:=;-\*#BR@aab-:=;#;