package com.company;

public class Main {

public static void main(String[] args) {

StrVal sv = new StrVal();

String table[][] = new String[12][9];

String grammar[][] = new String[6][2];

String follow[][] = new String[3][2];

String input = "i\*i+i";

for (int i = 0; i < 12; i++) {

for (int j = 0; j < 9; j++) {

table[i][j] = "#";

}

}

table[0][2] = "S04"; table[0][4] = "S05"; table[0][6] = "1"; table[0][7] = "2"; table[0][8] = "3";

table[1][0] = "S06"; table[1][5] = "ACCEPT";

table[2][0] = "R02"; table[2][1] = "S07"; table[2][3] = "R02"; table[2][5] = "R02";

table[3][0] = "R04"; table[3][1] = "R04"; table[3][3] = "R04"; table[3][5] = "R04";

table[4][2] = "S04"; table[4][4] = "S05"; table[4][6] = "6"; table[4][7] = "2"; table[4][8] = "3";

table[5][0] = "R06"; table[5][1] = "R06"; table[5][3] = "R06"; table[5][5] = "R06";

table[6][2] = "S04"; table[6][4] = "S05"; table[6][7] = "9"; table[6][8] = "3";

table[7][2] = "S04"; table[7][4] = "S05"; table[7][8] = "10";

table[8][0] = "S06"; table[8][3] = "S11";

table[9][0] = "R01"; table[9][1] = "S07"; table[9][3] = "R01"; table[9][5] = "R01";

table[10][0] = "R03"; table[10][1] = "R03"; table[10][3] = "R03"; table[10][5] = "R03";

table[11][0] = "R05"; table[11][1] = "R05"; table[11][3] = "R05"; table[11][5] = "R05";

grammar[0][0] = "E"; grammar[0][1] = "E+T";

grammar[1][0] = "E"; grammar[1][1] = "T";

grammar[2][0] = "T"; grammar[2][1] = "T\*F";

grammar[3][0] = "T"; grammar[3][1] = "F";

grammar[4][0] = "F"; grammar[4][1] = "(E)";

grammar[5][0] = "F"; grammar[5][1] = "i";

sv.validateStr(table, grammar, input);

}

}

StrVal

package com.company;

import java.util.Hashtable;

import java.util.Stack;

public class StrVal {

public StrVal(){}

public void validateStr(String[][] table, String[][] grammar, String input){

int countItr=0;

Stack buffer = new Stack();

Stack ist = new Stack();

Stack st = new Stack();

st.push("0");

Hashtable<String,Integer> ter=new Hashtable<String,Integer>();

ter.put("+", 0); ter.put("\*", 1); ter.put("(", 2); ter.put(")", 3); ter.put("i", 4); ter.put("$", 5);

ter.put("E", 6); ter.put("T", 7); ter.put("F", 8);

for (int i = 0; i < input.length(); i++) {

ist.push(String.valueOf(input.charAt(i)));

//System.out.print(ist.peek()+" ");

}

System.out.println("\nInitial Buffer: ");

buffer.push("$");

System.out.print(buffer.peek()+" ");

for (int i = 0; i < input.length(); i++) {

buffer.push(ist.pop());

System.out.print(buffer.peek()+" ");

}

System.out.println("\nIteration: "+ countItr);

System.out.println("Stack: "+ st.peek());

System.out.println("Buffer: " + buffer.peek());

do{

//str = table[Integer.parseInt((String)st.peek())][Integer.parseInt((String) ter.get((String)buffer.peek()))];

countItr++;

System.out.println("\nIteration: "+ countItr);

String str = table[Integer.parseInt((String)st.peek())][ter.get((String)buffer.peek())];

System.out.println("Action: "+ str);

if(str.equals("ACCEPT")){

System.out.println("Coming out of loop");

break;

}

if(String.valueOf(str.charAt(0)).equals("S")){

System.out.println("Adding to stack: ");

st.push(buffer.pop());

System.out.print(st.peek());

if(String.valueOf(str.charAt(1)).equals("0")){

st.push((String.valueOf(str.charAt(2))));

System.out.print(" "+st.peek());

}

else if(String.valueOf(str.charAt(1)).equals("1")){

String pushTemp = String.valueOf(str.charAt(0)) + String.valueOf(str.charAt(1));

st.push(pushTemp);

System.out.print(" "+st.peek());

}

}

else if(String.valueOf(str.charAt(0)).equals("R")){

if(String.valueOf(str.charAt(1)).equals("0")){

int num = Integer.parseInt(String.valueOf(str.charAt(2)));

String temp = grammar[num-1][1];

int count = 2\*(temp.length());

System.out.println("Deleting from stack: ");

while(count!=0){

System.out.print(" "+st.peek() );

st.pop();

count--;

}

System.out.println("\n");

System.out.print(st.peek());

String temp2 = (String) st.peek();

st.push(grammar[num-1][0]);

System.out.print(" "+st.peek());

st.push(table[Integer.parseInt(temp2)][ter.get((String)st.peek())]);

System.out.print(" "+st.peek());

}

// else if(String.valueOf(str.charAt(1)).equals("1")){

// int num2 = Integer.parseInt(String.valueOf(str.charAt(0)) + String.valueOf(str.charAt(1)));

// String temp = grammar[num2-1][1];

// int count = 2\*(temp.length());

// while(count!=0){

// st.pop();

// count--;

// }

// String temp2 = (String) st.peek();

// st.push(grammar[num2-1][0]);

// st.push(table[Integer.parseInt(temp2)][ter.get((String)st.peek())]);

// }

}

System.out.println("\nBuffer: "+ buffer.peek());

}while((Integer.parseInt((String)st.peek()) != 1 || (buffer.size()!=1 && !((String)buffer.peek()).equals("$"))) || st.empty());

//}while(!String.valueOf(str.charAt(0)).equals("A"));

System.out.println("String is valid!");

System.out.println("The stack contents are: ");

while(!st.empty()){

System.out.print(st.pop() + " ");

}

}

}