**Experiment No. 9**

**Aim: Generate target code for the optimized code, considering the target machines to be x86**

Code:

package codegenerator;

import java.util.\*;

public class CodeGenerator {

static String kw(char a) {

if(a=='\*') return "MULF";

if(a=='/') return "DIVF";

if(a=='+') return "ADDF";

if(a=='-') return "SUBF";

return "ERR";

}

static int prior(char a) {

if(a=='\*' || a=='/')

return 1;

if(a=='+' || a=='-')

return 2;

return 3;

}

public static void main(String args[]) {

String sym="",token,id="",va[],t="",val="";

List<String> var = new ArrayList<String>();

char sy[],sm=' ',in,sep[]={'=','/','\*','+','-','%','(',')','{','}'};

int tlen,n=0,ns,tp;

boolean flag,sd[];

Scanner s = new Scanner(System.in);

System.out.println("Enter token :");

token=s.next();

tlen = token.length();

for(int i=tlen-1;i>=0;i--) {

flag=false;

in=token.charAt(i);

for(int j=0;j<10;j++) {

if(in==sep[j]) {

if( in==')'|| in=='}') {}

else if( in=='('|| in=='{' ) {}

else {

var.add(id);

if(in!='=')

sym =sym + in;

id="";

}

flag=true;

}

if(flag)

break;

}

if(flag)

continue;

else {

id= in + id;

} }

ns=sym.length();

va=var.toArray(new String[var.size()]);

sd=new boolean[ns];

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

System.out.println("MOVF "+ va[i] + " ," + "R"+n + ";");

va[i]="R"+n++; }

tp=0;

n=0;

while(n<ns) {

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

in=sym.charAt(i);

if(prior(sm)>prior(in) && sd[i]==false) {

sm=in;

tp=i;

} }

sd[tp]=true;

System.out.println(kw(sm) + " " + va[tp+1] + " ," + va[tp] + ";");

sm=' ';

va[tp+1]=va[tp];

n++;

}

System.out.println("MOVF " + va[tp] + " ," + id);

} }

**Output:**

Enter token :

a=b+c\*60

MOVF 60 ,R0;

MOVF c ,R1;

MOVF b ,R2;

MULF R1 ,R0;

ADDF R2 ,R0;

MOVF R0 ,a