15/04/2022

Experiment 11 : Representation of intermediate code : Quadruples, Triples, three address code.

Aim: A program to implement intermediate code generation using quadruples, Triples and three address code.

Algorithm:

1. Start
2. Input is in the form of a sequence of three-adress statements. For each three adress code of the form a := b op c perform the following:

i) Invoke a function to find the location of L where result of the computation b op c should be stored.

ii) Consult the address description for y to determine y'. If the value of y is not already in L then generate the instruction MOV y', L to place a copy of y in L.

iii) Generate the instruction OP z', L where z' is used to show the current location of z. If z is in both then prefer a register to a memory location. Update the address descriptor of x to indicate that x is in location L. If x is in L then update its descriptor and remove x from all other descriptors.

iv) If the current value of y or z have no next uses or not live on exit from the block or in register then alter the register descriptor to indicate that after execution of $x := y \text{ op } z$ those registers will no longer contain y or z

MANUAL WORKING

Input:

$$a = b + c - d$$

Intermediate code:

$$t_1 = b + c$$
$$t_2 = t_1 - d$$
$$a = t_2$$

Sashrika Surya
RA1911027010092
Section N2
Date: 15/4/2022

Lab11: Representation of Intermediate Code - Quadruples, triples, indirect triple

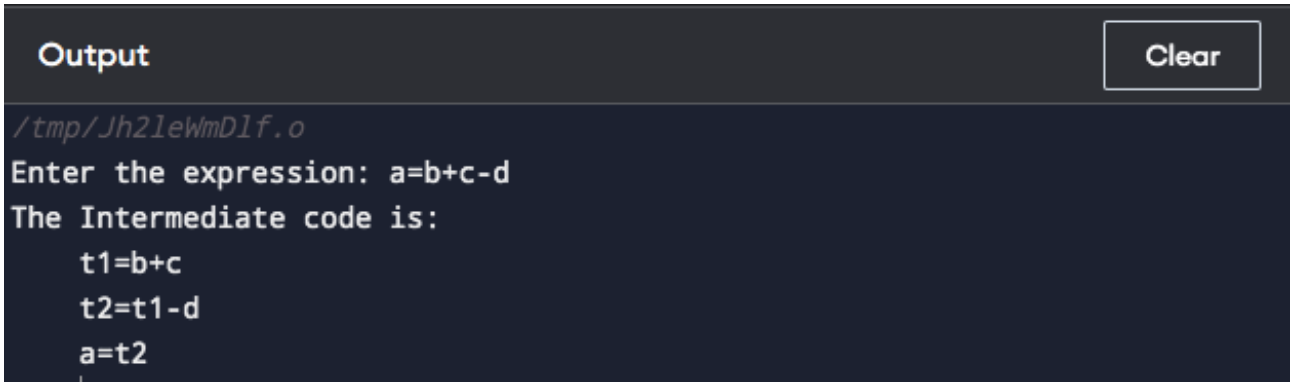
Code:

```
#include<stdio.h>
#include<ctype.h>
#include<stdlib.h>
#include<string.h>
void small();
void dove(int i);
int p[5]={0,1,2,3,4},c=1,i,k,l,m,pi;
char sw[5]={'=','-','+','/','*'},j[20],a[5],b[5],ch[2];
int main()
{
printf("Enter the expression: ");
scanf("%s",j);
printf("\tThe Intermediate code is:\n");
small();
return 0;
}
void dove(int i)
{
a[0]=b[0]='\0';
if(!isdigit(j[i+2])&&!isdigit(j[i-2]))
{
a[0]=j[i-1];
b[0]=j[i+1];
}
if(isdigit(j[i+2])){
a[0]=j[i-1];
b[0]='t';
b[1]=j[i+2];
}
if(isdigit(j[i-2]))
{
b[0]=j[i+1];
a[0]='t';
```

```
a[1]=j[i-2];
b[1]='\0';
}
if(isdigit(j[i+2]) &&isdigit(j[i-2]))
{
a[0]='t';
b[0]='t';
a[1]=j[i-2];
b[1]=j[i+2];
sprintf(ch,"%d",c);
j[i+2]=j[i-2]=ch[0];
}
if(j[i]=='*')
printf("\tt%d=%s*%s\n",c,a,b);
if(j[i]=='/')
printf("\tt%d=%s/%s\n",c,a,b);
if(j[i]=='+')
printf("\tt%d=%s+%s\n",c,a,b);if(j[i]=='-')
printf("\tt%d=%s-%s\n",c,a,b);
if(j[i]=='=')
printf("\t%c=t%d",j[i-1],--c);
sprintf(ch,"%d",c);
j[i]=ch[0];
c++;
small();
}
void small()
{
pi=0;l=0;
for(i=0;i<strlen(j);i++)
{
for(m=0;m<5;m++)
if(j[i]==sw[m])
if(pi<=p[m])
{
pi=p[m];
 l=1;
 k=i;
} }
if(l==1)
dove(k);
else
exit(0);

}
```

Output:



Result:

Hence, Intermediate Code was represented quadruples, triples and three address code.