Experiment Number 11

Aim: Intermediate code generation – Quadruple, Triple, Indirect triple.

Algorithm:

Step 1: Start

Step 2: Read code.

Step 3: Replace expressions by temp variables.

Step 4: Join the codes logically.

Step 5: Print respective output.

Step 6: Stop.

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],s[20];
void main()
{
printf("Enter the expression:");
scanf("%s",j);
int x=0,f=0;
char k[20],y=0;
for(int i=0;i<strlen(j);i++){
   if(j[i]!='(' && f!=1)
   s[x++]=j[i];
   else if(j[i]=='('){
      f=1;
   }
   else if(j[i]==')'){
      f=0;
      s[x++]='$';
   }
   else if(f==1)
   {
      k[y++]=j[i];
   }
```

```
}
//printf("%s",s);
printf("\tThe Intermediate code is:\n");
printf("\tt$=%s\n",k);
strcpy(j,s);
small();
}
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:

```
Enter the expression:a=b+c*d
        The Intermediate code is:
        t$=
        t1=c*d
        t2=b+t1
        a=t2
```

**t$ indicates bracketed operations, in this input there aren't any, therefore empty.

Result: Thus, Intermediate code generation – Quadruple, Triple, Indirect triple codes implemented successfully.