

Aim:-

TO construct a Simple Code Generator using C language.

Algorithm:-

The algorithm takes as input a sequence of three-address statements constituting a basic block. For each three-address statement of the form  $x := y \text{ op } z$ , perform the following actions:

1. Invoke a function `getreg` to determine the location  $L$  where the result of the computation  $y \text{ op } z$  should be stored.
2. Consult the address descriptor for  $y$  to determine  $y'$ , the current location of  $y$ . Prefer the register for  $y'$  if the value of  $y$  is currently both in memory and a register. If the value of  $y$  is not already in  $L$ , generate the instruction `MOV  $y'$ ,  $L$`  to place a copy of  $y$  in  $L$ .
3. Generate the instruction `OP  $z'$ ,  $L$`  where  $z'$  is a current location of  $z$ . Prefer a register to a memory location if  $z$  is in both. Update the address descriptor of  $x$  to indicate that  $x$  is in location  $L$ . If  $x$  is in  $L$ , update its descriptor and remove  $x$  from all other descriptors.
4. If the current values of  $y$  or  $z$  have no next uses, are not live on exit from the block, and are in registers, alter the register descriptor to indicate that, after execution of  $x := y \text{ op } z$ , those registers will no longer contain  $y$  or  $z$ .

Source code:-

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
char op[2],arg1[5],arg2[5],result[5];
void main()
{
    FILE *fp1,*fp2;
    fp1=fopen("input.txt","r");
    fp2=fopen("output.txt","w");
    while(!feof(fp1))
    {
        fscanf(fp1,"%s%s%s%s",op,arg1,arg2,result);
        if(strcmp(op,"+")==0)
        {
            fprintf(fp2,"\nMOV R0,%s",arg1);
            fprintf(fp2,"\nADD R0,%s",arg2);
        }
    }
}
```

```

    fprintf(fp2, "\nMOV %s, R0", result);
}
if(strcmp(op, "*")==0)
{
    fprintf(fp2, "\nMOV R0, %s", arg1);
    fprintf(fp2, "\nMUL R0, %s", arg2);
    fprintf(fp2, "\nMOV %s, R0", result);
}
if(strcmp(op, "-")==0)
{
    fprintf(fp2, "\nMOV R0, %s", arg1);
    fprintf(fp2, "\nSUB R0, %s", arg2);
    fprintf(fp2, "\nMOV %s, R0", result);
}
if(strcmp(op, "/")==0)
{
    fprintf(fp2, "\nMOV R0, %s", arg1);
    fprintf(fp2, "\nDIV R0, %s", arg2);
    fprintf(fp2, "\nMOV %s, R0", result);
}
if(strcmp(op, "=")==0)
{
    fprintf(fp2, "\nMOV R0, %s", arg1);
    fprintf(fp2, "\nMOV %s, R0", result);
}
}
fclose(fp1);
fclose(fp2);
getch();
}

```

Input.txt:-

+ a b t1

\* c d t2

- t1 t2 t

= t ? x

### Manual Calculation:-

Let the Expressions be :-

$+ a \ b \ t_1$

$\times c \ d \ t_2$

$- t_1 \ t_2 \ t$

$= t \ ? \ x$

First assign  $a$  to  $R_0$

MOV  $R_0, a$

Add  $b$  to  $a$  i.e.  $R_0$

ADD  $R_0, b$

Assign  $a+b$  value to  $t_1$

MOV  $t_1, R_0$

Assign  $c$  to  $R_0$

MOV  $R_0, c$

multiply with  $d$

MUL  $R_0, d$

Assign  $c \times d$  to  $t_2$

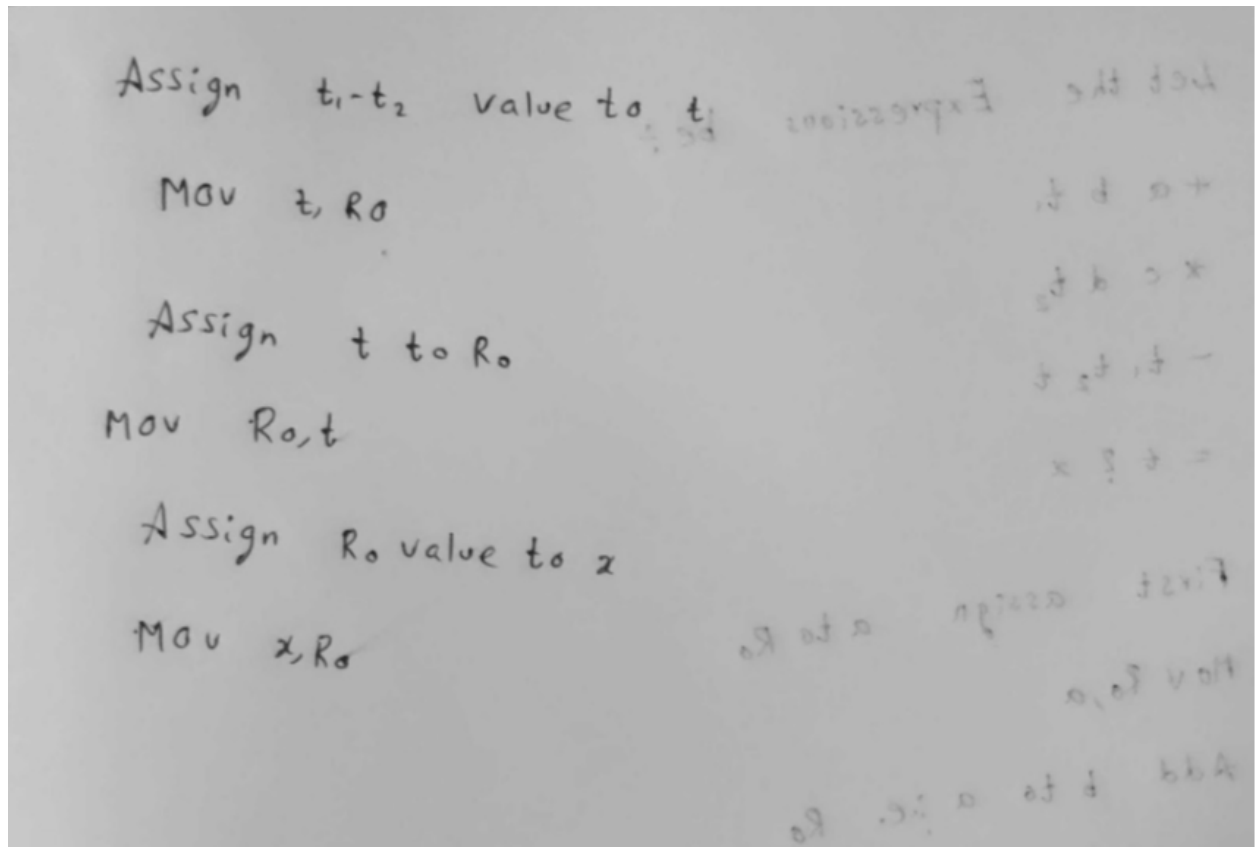
MOV  $t_2, R_0$

Assign  $t_1$  to  $R_0$

MOV  $R_0, t_1$

Subtract  $t_2$  from  $t_1$  i.e.  $R_0$

SUB  $R_0, t_2$



Output:-

```

1
2 MOV R0,a
3 ADD R0,b
4 MOV t1,R0
5 MOV R0,c
6 MUL R0,d
7 MOV t2,R0
8 MOV R0,t1
9 SUB R0,t2
10 MOV t,R0
11 MOV R0,t
12 MOV x,R0

```

Result:-

Simple Code Generator has been successfully constructed using c program.