PRACTICAL-8

AIM:

Implementation of code generator.

PROGRAM CODE:

```
#include<stdio.h>
#include<string.h>
struct table{
char op1[2];
char op2[2];
char opr[2];
char res[2];
}tbl[100];

void add(char *res,char *op1, char *op2,char *opr)
{
   FILE *ft;
   char string[20];
   char sym[100];
   ft=fopen("result.asm","a+");
   if(ft==NULL)
      ft=fopen("result.asm","w");
   printf("\nUpdating Assembly Code for the Input File : File : Result.asm ; Status [ok]\n");
   //sleep(2);
   strcpy(string,"mov r0,");
   strcat(string,op1);
   if(strcmp(opr,"&")==0)
   {
      //do nothing
   }
   else
   {
      strcat(string,"\nmov r1,");
      strcat(string,op2);
   }
   fputs(string,ft);
   if(strcmp(opr,"+")==0)
      strcpy(string,"\nadd r0,r1\n");
```

```
    else if(strcmp(opr,"-")==0)
        strcpy(string,"\nsub r0,r1\n");
    else if(strcmp(opr,"/")==0)
        strcpy(string,"\ndiv r0,r1\n");
    else if(strcmp(opr,"*")==0)
        strcpy(string,"\nmul r0,r1\n");
    else if(strcmp(opr,"&")==0)
        strcpy(string,"\n");
    else
        strcpy(string,"\noperation r0,r1\n");
    fputs(string,ft);
    strcpy(string,"mov ");
    strcat(string,res);
    strcat(string,", r0\n");
    fputs(string,ft);
    fclose(ft);
    string[0]='\0';
    sym[0]='\0';
}
main()
{
    int res,op1,op2,i,j,opr;
    FILE *fp;
    char filename[50];
    char s,s1[10];
    system("clear");
    remove("result.asm");
    remove("result.sym");
    res=0;op1=0;op2=0;i=0;j=0;opr=0;
    printf("\nPARTH PATEL\n19DCS098\n");
    printf("\n Enter the Input Filename with no white spaces:");
    scanf("%s",filename);
    fp=fopen(filename,"r");
    if(fp==NULL)
    {
        printf("\n cannot open the input file !\n");
        return(0);
    }
    else
    {
        while(!feof(fp))
        {
```

```
s=fgetc(fp);
   if(s=='=')
   {
      res=1;
      op1=op2=opr=0;
      s1[j]='\0';
      strcpy(tbl[i].res,s1);
      j=0;
   }
   else if(s=='+'||s=='-'||s=='*'||s=='/')
   {
      op1=1;
      opr=1;
      s1[j]='\0';
      tbl[i].opr[0]=s;
      tbl[i].opr[1]='\0';
      strcpy(tbl[i].op1,s1);
      j=0;
   }
   else if(s==';')
   {

      if(opr)     // for 3 operand format ex: a=b+c;
      {
         op2=1;
         s1[j]='\0';
         strcpy(tbl[i].op2,s1);
      }
      else if(!opr)   // for 2 operand format ex: d=a;
      {
         op1=1;
         op2=0;
         s1[j]='\0';
         strcpy(tbl[i].op1,s1);
         strcpy(tbl[i].op2,"&");
         strcpy(tbl[i].opr,"&");
      }
      add(tbl[i].res,tbl[i].op1,tbl[i].op2,tbl[i].opr);
      i++;
      j=0;
      opr=op1=op2=res=0;
   }
```

```
            else
            {
               s1[j]=s;
               j++;
            }
      }
   system("clear");
}
   return 0;
}
```

INPUT:

a=b+c;

d=n+s;

p=q;

OUTPUT:

```
mov r0,b
mov r1,c
add r0,r1
mov a, r0
mov r0,n
mov r1,s
add r0,r1
mov
d, r0
mov r0,q
mov
p, r0
```