**AIM:**  
             To write a "C" program for the implementation of a single pass assembler in CS1207 - System Software Lab.  
  
**SOURCE CODE:**  
  
#include<stdio.h>  
#include<conio.h>  
#include<string.h>  
void main()  
{  
 char opcode[10],operand[10],label[10],a[10],ad[10],symbol[10],ch; char code[10][10],code1[10][10]={"33","44","53","57"};  
 char mnemonic[10][10]={"START","LDA","STA","LDCH","STCH","END"};  
 char mnemonic1[10][10]={"LDA","STA","LDCH","STCH"};  
 int locctr,start,length,i=0,j=0,k,l=0;  
 int st,diff,address,add,len,actual\_len,finaddr,prevaddr;  
 FILE \*fp1,\*fp2,\*fp3,\*fp4,\*fp5,\*fp6,\*fp7;  
 clrscr();  
 fp1=fopen("INPUT.DAT","r");  
 fp2=fopen("SYMTAB.DAT","w");  
 fp3=fopen("INETERMED.DAT","w");  
 fscanf(fp1,"%s%s%s",label,opcode,operand);  
 if(strcmp(opcode,"START")==0)  
  {  
   start=atoi(operand);  
   locctr=start;  
   fprintf(fp3,"%s\t%s\t%s\n",label,opcode,operand);  
   fscanf(fp1,"%s%s%s",label,opcode,operand);  
  }  
 else  
  locctr=0;  
 while(strcmp(opcode,"END")!=0)  
  {  
   fprintf(fp3,"%d",locctr);  
   if(strcmp(label,"\*\*")!=0)  
     fprintf(fp2,"%s\t%d\n",label,locctr);  
   strcpy(code[i],mnemonic[j]);  
   while(strcmp(mnemonic[j],"END")!=0)  
    {  
     if(strcmp(opcode,mnemonic[j])==0)  
     {  
      locctr+=3;  
      break;  
     }  
     strcpy(code[i],mnemonic[j]);  
     j++;  
    }  
   if(strcmp(opcode,"WORD")==0)  
    locctr+=3;  
   else if(strcmp(opcode,"RESW")==0)  
    locctr+=(3\*(atoi(operand)));  
   else if(strcmp(opcode,"RESB")==0)  
    locctr+=(atoi(operand));  
   else if(strcmp(opcode,"BYTE")==0)  
    ++locctr;  
   fprintf(fp3,"\t%s\t%s\t%s\n",label,opcode,operand);  
   fscanf(fp1,"%s%s%s",label,opcode,operand);  
  }  
  fprintf(fp3,"%d\t%s\t%s\t%s\n",locctr,label,opcode,operand);  
  length=locctr-start;  
  fcloseall();  
  printf("\n\nThe contents of Input file:\n\n");  
  fp1=fopen("INPUT.DAT","r");  
  ch=fgetc(fp1);  
  while(ch!=EOF)  
   {  
    printf("%c",ch);  
    ch=fgetc(fp1);  
   }  
  printf("\n\nLength of the input program is %d.",length);  
  printf("\n\nThe contents of Symbol Table:\n\n");  
  fp2=fopen("SYMTAB.DAT","r");  
  ch=fgetc(fp2);  
  while(ch!=EOF)  
   {  
    printf("%c",ch);  
    ch=fgetc(fp2);  
   }  
  fcloseall();  
  fp4=fopen("ASSMLIST.DAT","w");  
  fp5=fopen("SYMTAB.DAT","r");  
  fp6=fopen("INTERMED.DAT","r");  
  fp7=fopen("OBJCODE.DAT","w");  
  fscanf(fp6,"%s%s%s",label,opcode,operand);  
  while(strcmp(opcode,"END")!=0)  
  {  
   prevaddr=address;  
   fscanf(fp6,"%d%s%s%s",&address,label,opcode,operand);  
  }  
  finaddr=address;  
  fclose(fp6);  
  fp6=fopen("INTERMED.DAT","r");  
  fscanf(fp6,"%s%s%s",label,opcode,operand);  
  if(strcmp(opcode,"START")==0)  
  {  
   fprintf(fp4,"\t%s\t%s\t%s\n",label,opcode,operand);  
   fprintf(fp7,"H^%s^00%s^00%d\n",label,operand,finaddr);  
   fscanf(fp6,"%d%s%s%s",&address,label,opcode,operand);  
   st=address;  
   diff=prevaddr-st;  
   fprintf(fp7,"T^00%d^%d",address,diff);  
  }  
  while(strcmp(opcode,"END")!=0)  
  {  
   if(strcmp(opcode,"BYTE")==0)  
   {  
    fprintf(fp4,"%d\t%s\t%s\t%s\t",address,label,opcode,operand);  
    len=strlen(operand);  
    actual\_len=len-3;  
    fprintf(fp7,"^");  
    for(k=2;k<(actual\_len+2);k++)  
    {  
     itoa(operand[k],ad,16);  
     fprintf(fp4,"%s",ad);  
     fprintf(fp7,"%s",ad);  
    }  
    fprintf(fp4,"\n");  
   }  
   else if(strcmp(opcode,"WORD")==0)  
   {  
    len=strlen(operand);  
    itoa(atoi(operand),a,10);  
    fprintf(fp4,"%d\t%s\t%s\t%s\t00000%s\n",address,label,opcode,operand,a);  
    fprintf(fp7,"^00000%s",a);  
   }  
   else if((strcmp(opcode,"RESB")==0)||(strcmp(opcode,"RESW")==0))  
    fprintf(fp4,"%d\t%s\t%s\t%s\n",address,label,opcode,operand);  
   else  
   {  
    while(strcmp(opcode,mnemonic1[l])!=0)  
     l++;  
    if(strcmp(operand,"COPY")==0)  
     fprintf(fp4,"%d\t%s\t%s\t%s\t%s0000\n",address,label,opcode,operand,code1[l]);  
    else  
    {  
     rewind(fp5);  
     fscanf(fp5,"%s%d",symbol,&add);  
      while(strcmp(operand,symbol)!=0)  
       fscanf(fp5,"%s%d",symbol,&add);  
     fprintf(fp4,"%d\t%s\t%s\t%s\t%s%d\n",address,label,opcode,operand,code1[l],add);  
     fprintf(fp7,"^%s%d",code1[l],add);  
    }  
   }  
   fscanf(fp6,"%d%s%s%s",&address,label,opcode,operand);  
  }  
  fprintf(fp4,"%d\t%s\t%s\t%s\n",address,label,opcode,operand);  
  fprintf(fp7,"\nE^00%d",st);  
  printf("\nObject Program has been generated.");  
  fcloseall();  
  printf("\n\nObject Program:\n\n");  
  fp7=fopen("OBJCODE.DAT","r");  
  ch=fgetc(fp7);  
  while(ch!=EOF)  
  {  
   printf("%c",ch);  
   ch=fgetc(fp7);  
  }  
  fcloseall();  
  getch();  
}

**INPUT FILE:**

**INPUT.DAT**  
COPY START 2000  
\*\* LDA FIVE  
\*\* STA ALPHA  
\*\* LDCH CHARZ  
\*\* STCH C1  
ALPHA RESW 1  
FIVE WORD 5  
CHARZ BYTE C'EOF'  
C1 RESB 1  
\*\* END \*\*  
  
**OUTPUT:**

[](http://1.bp.blogspot.com/_V648gqhagYA/TTAWYprHlrI/AAAAAAAAAn4/s_-JtNOuiTg/s1600/single-pass-assembler-implementation-c-source-code-cs1207-system-software-lab.JPG)