Pgm.No.11

PASS TWO OF TWO PASS ASSEMBLER

AIM

Write a program to implement pass one of two pass assembler

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
    {
char opcode[20],operand[20],symbol[20],label[20],code[20],mnemonic[25], character,
add[20],objectcode[20];
int flag,flag1,locctr,location,loc;
FILE *fp1,*fp2,*fp3,*fp4;

fp1=fopen("out3.txt","r"); fp2=fopen("twoout.txt","w");
fp3=fopen("opcode.txt","r"); fp4=fopen("sym1.txt","r");
fscanf(fp1,"%s%s%s",label,opcode,operand);
if(strcmp(opcode,"START")==0)
{ fprintf(fp2,"%s\t%s\t%s\n",label,opcode,operand);
fscanf(fp1,"%d%s%s%s",&locctr,label,opcode,operand);
}
while(strcmp(opcode,"END")!=0)
{ flag=0;
fscanf(fp3,"%s%s",code,mnemonic);
while(strcmp(code,"END")!=0)
{ if((strcmp(opcode,code)==0) && (strcmp(mnemonic,"*"))!=0)
{ flag=1;
break;
}
fscanf(fp3,"%s%s",code,mnemonic);


}
if(flag==1)
{ flag1=0; rewind(fp4);
while(!feof(fp4))
```

```
{
fscanf(fp4,"%s%d",symbol,&loc);
if(strcmp(symbol,operand)==0)
{
flag1=1; break;
} }
if(flag1==1)
{
sprintf(add,"%d",loc);
strcpy(objectcode,strcat(mnemonic,add));
} }
else if(strcmp(opcode,"BYTE")==0 || strcmp(opcode,"WORD")==0)
{
if((operand[0]=='C') || (operand[0]=='X'))
{
character=operand[2];
sprintf(add,"%d",character);
strcpy(objectcode,add);
}
else
{
strcpy(objectcode,add);
} }
else
strcpy(objectcode,"\0");
fprintf(fp2,"%s\t%s\t%s\t%d\t%s\n",label,opcode,operand,locctr,objectcode);
fscanf(fp1,"%d%s%s%s",&locctr,label,opcode,operand);
}
fprintf(fp2,"%s\t%s\t%s\t%d\n",label,opcode,operand,locctr);
fclose(fp1);
fclose(fp2);
fclose(fp3);
 fclose(fp4);
}
```

INPUT FILES

opcode.txt

```
START *
LDA 03
STA 0F
LDCH 53
STCH 57
END +
```

out3.txt

```
** START 2000
2000 ** LDA FIVE
2003 ** STA ALPHA
2006 ** LDCH CHARZ
2009 ** STCH C1
2012 ALPHA RESW 1
2015 FIVE WORD 5
2018 CHARZ BYTE C'Z'
2019 C1 RESB 1
2020 ** END **
```

sym1.txt

```
2012 ALPHA
2015 FIVE
2018 CHARZ
2019 C1
```

OUTPUT FILES

twoout.txt

```
**      START       2000
**      LDA   FIVE  2000   032018
**      STA   ALPHA       2003   0F2015
**      LDCH CHARZ        2006   532019
**      STCH  C1    2009   572019
ALPHA       RESW        1       2012
FIVE   WORD         5       2015   2019
CHARZ       BYTE C'Z'   2018   90
C1      RESB  1       2019
**      END   **      2020
```


1. Define the basic functions of assembler.
* Translating mnemonic operation codes to their machine language equivalents.
* Assigning machine addresses to symbolic labels used by the programmer.

2. What is meant by assembler directives? Give example.
These are the statements that are not translated into machine instructions, but they provide instructions to assembler itself.
example START,END,BYTE,WORD,RESW and RESB.

3. What are forward references?
It is a reference to a label that is defined later in a program.
Consider the statement

```
10 1000 STL RETADR
. . . .
. . . .
80 1036 RETADR RESW 1
```

The first instruction contains a forward reference RETADR. If we attempt to translate the program line by line, we will unable to process the statement in line10 because we do not know the address that will be assigned to RETADR .The address is assigned later(in line 80) in the program.

4. What are the three different records used in object program?
The header record, text record and the end record are the three different records used in object program.
The header record contains the program name, starting address and length of the program.
Text record contains the translated instructions and data of the program.
End record marks the end of the object program and specifies the address in the program where