Implement Pass1 of Two Pass Assembler

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main()
{
    FILE *optab,*input,*symtab,*inter,*len;
    int locctr,start=0;
    char mnemonic[15],opcode[25],operand[25],label[25],code[25];
    optab=fopen("OPTAB.txt","r");
    input=fopen("INPUT.txt","r");
    symtab=fopen("SYMTAB.txt","w");
    inter=fopen("INTER.txt","w");
    len=fopen("LEN.txt","w");
    if(optab==NULL||input==NULL||symtab==NULL||inter==NULL)
        printf("One File is not open");

    fscanf(input,"%s\t%s\t%s",label,opcode,operand);
    if(strcmp(opcode,"START")==0)
    {
        start=atoi(operand);
        locctr=start;
        printf("START IS %d\n",start);
        fprintf(inter,"%s\t%s\t%s\n",label,opcode,operand);
    }
    else
    {
        locctr=0;
    }
    fscanf(input,"%s\t%s\t%s",label,opcode,operand);
    while(strcmp(opcode,"END")!=0)
    {
        fprintf(inter,"%d\t",locctr);
        if(strcmp(label,"**")!=0)
        {
            fprintf(symtab,"%s\t%d\n",label,locctr);
        }
        fscanf(optab,"%s\t%s",code,mnemonic);
        while(!feof(optab)) //or while(strcmp(code,"END")!=0)
        {
```

```
        if(strcmp(code,opcode)==0)
        {
            locctr+=3;
            rewind(optab);
            break;
        }
        fscanf(optab,"%s\t%s",code,mnemonic);
    }
    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)
    {

        if (operand[0]=='C')
        {
            locctr+=(strlen(operand)-3);
        }
        else if(operand[0]=='X')
        {
            printf("\n##1##");
            locctr++;
        }
    }

    fprintf(inter,"%s\t%s\t%s\n",label,opcode,operand);
    fscanf(input,"%s\t%s\t%s",label,opcode,operand);
}
fprintf(inter,"%s\t%s\t%s\n",label,opcode,operand);
printf("Length of program is %d",locctr-start);
```

```
    fprintf(len,"%d",locctr-start);
    fscanf(len,"%d",locctr-start);
    fclose(len);
    fclose(optab);
    fclose(inter);
    fclose(symtab);
    fclose(input);
}
```

INPUT

OPTAB.txt

```
LDA     00
MUL     20
STA     0C
LDCH    50
STCH    54
```

INPUT.txt

```
MUL     START   1000
**      LDA     ALPHA
**      MUL     BETA
**      STA     GAMMA
ALPHA   WORD    2
BETA    WORD    4
GAMMA   RESW    1
**      END     **
```

OUTPUT

mulSymtab.txt

```
ALPHA     1009
BETA      1012
GAMMA     1015
```

INTERMEDIATE.txt

```
MUL START      1000
```

```
1000 **   LDA  ALPHA
1003 **   MUL  BETA
1006 **   STA  GAMMA
1009 ALPHA     WORD      2
1012 BETA      WORD      4
1015 GAMMA     RESW      1
**   END **
```

Implement A Single Pass Assembler

Onepass.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char *my_itoa(int num, char *str)
{
    if (str == NULL)
    {
        return NULL;
    }
    sprintf(str, "%d", num);
    return str;
}

void main()
{
    FILE *f1, *f2, *f3, *f4, *f5;
    int lc, sa, i = 0, j = 0, m[10], pgmlen, len, k, len1, l = 0;
    char name[10], opnd[10], la[10], mne[10], s1[10], mne1[10], opnd1[10];
    char lcs[10], ms[10];
    char sym[10], symaddr[10], obj1[10], obj2[10], s2[10], q[10], s3[10];
    f1 = fopen("INPUT.txt", "r");
    f2 = fopen("OPTAB.txt", "r");
    f3 = fopen("symtab.txt", "w+");
    f4 = fopen("symtab1.txt", "w+");
    f5 = fopen("output.txt", "w+");
    fscanf(f1, "%s%s%s", la, mne, opnd);
    if (strcmp(mne, "START") == 0)
    {
        sa = atoi(opnd);
        strcpy(name, la);
        lc = sa;
    }
    strcpy(s1, "*");
    fscanf(f1, "%s%s%s", la, mne, opnd);
    while (strcmp(mne, "END") != 0)
    {
        if (strcmp(la, "-") == 0)
```

```
{
    fscanf(f2, "%s%s", mne1, opnd1);
    while (!feof(f2))
    {
        if (strcmp(mne1, mne) == 0)
        {
            m[i] = lc + 1;
            fprintf(f3, "%s\t%s\n", opnd, s1);
            fprintf(f5, "%s\t0000\n", opnd1);
            lc = lc + 3;
            i = i + 1;
            break;
        }
        else
            fscanf(f2, "%s%s", mne1, opnd1);
    }
}
else
{
    fseek(f3, SEEK_SET, 0);
    fscanf(f3, "%s%s", sym, symaddr);
    while (!feof(f3))
    {
        if (strcmp(sym, la) == 0)
        {
            my_itoa(lc, lcs);
            fprintf(f4, "%s\t%s\n", la, lcs);
            my_itoa(m[j], ms);
            j = j + 1;
            fprintf(f5, "%s\t%s\n", ms, lcs);
            i = i + 1;
            break;
        }
        else
            fscanf(f3, "%s%s", sym, symaddr);
    }
    if (strcmp(mne, "RESW") == 0)
        lc = lc + 3 * atoi(opnd);
    else if (strcmp(mne, "BYTE") == 0)
    {
        strcpy(s2, "-");
```

```
                len = strlen(opnd);
                lc = lc + len - 2;
                for (k = 2; k < len; k++)
                {
                    q[l] = opnd[k];
                    l = l + 1;
                }
                fprintf(f5, "%s\t%s\n", q, s2);
                break;
            }
            else if (strcmp(mne, "RESB") == 0)
                lc = lc + atoi(opnd);
            else if (strcmp(mne, "WORD") == 0)
            {
                strcpy(s3, "#");
                lc = lc + 3;
                fprintf(f5, "%s\t%s\n", opnd, s3);
                break;
            }
        }

        fseek(f2, SEEK_SET, 0);
        fscanf(f1, "%s%s%s", la, mne, opnd);
    }
    fseek(f5, SEEK_SET, 0);
    pgmlen = lc - sa;
    printf("H^%s^%d^0%x\n", name, sa, pgmlen);
    printf("T^");
    printf("00%d^0%x", sa, pgmlen);
    fscanf(f5, "%s%s", obj1, obj2);
    while (!feof(f5))
    {
        if (strcmp(obj2, "0000") == 0)
            printf("^%s%s", obj1, obj2);
        else if (strcmp(obj2, "-") == 0)
        {
            printf("^");
            len1 = strlen(obj1);
            for (k = 0; k < len1; k++)
                printf("%d", obj1[k]);
        }
        else if (strcmp(obj2, "#") == 0)
        {
            printf("^");
            printf("%s", obj1);
        }
        fscanf(f5, "%s%s", obj1, obj2);
    }
    fseek(f5, SEEK_SET, 0);
    fscanf(f5, "%s%s", obj1, obj2);
    while (!feof(f5))
    {
        if (strcmp(obj2, "0000") != 0)
        {
            if (strcmp(obj2, "-") != 0)
            {
                if (strcmp(obj2, "#") != 0)
                {
                    printf("\n");
                    printf("T^%s^02^%s", obj1, obj2);
                }
            }
        }
        fscanf(f5, "%s%s", obj1, obj2);
    }
    printf("\nE^00%d", sa);
}
```

INPUT

INPUT.txt

```
MUL     START   1000
-       LDA     ALPHA
-       MUL     BETA
-       STA     GAMMA
ALPHA   WORD    2
BETA    WORD    4
GAMMA   RESW    1
--      END     **
```

OPTAB.txt

```
LDA     00
```

```
LDX     04
LDT     74
ADD      18
SUB      1C
MUL      20
DIV     24
```

OUTPUT

Output.txt

```
00                                    0000
20                                    0000
1001                                  1006
2                                     #
```

symtab.txt

```
ALPHA                                 *
BETA                                  *
```

Symtab1.txt

```
ALPHA                                 1006
```

```
H^MUL^1000^09
T^001000^09^000000^200000^2
T^1001^02^1006
E^001000
```

Implementation a Two Pass Macroprocessor

Pass1.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void main()
{
    FILE *f1, *f2, *f3;
    char mne[20], opnd[20], la[20];
    f1 = fopen("input.txt", "r");
    f2 = fopen("namtab.txt", "w+");
    f3 = fopen("deftab.txt", "w+");
    fscanf(f1, "%s%s%s", la, mne, opnd);
    while (strcmp(mne, "MEND") != 0)
    {
        if (strcmp(mne, "MACRO") == 0)
        {
            fprintf(f2, "%s\n", la);
            fprintf(f3, "%s\t%s\n", la, opnd);
        }
        else
            fprintf(f3, "%s\t%s\n", mne, opnd);
        fscanf(f1, "%s%s%s", la, mne, opnd);
    }
    fprintf(f3, "%s", mne);
    fclose(f1);
    fclose(f2);
    fclose(f3);
    printf("\nPass 1 of 2 pass macroprocessor is successful.");
}
```

Input.txt

```
EX1    MACRO   &A,&B
-      LDA     &A
-      STA     &B
-      MEND    -
SAMPLE  START   1000
-      EX1     N1,N2
N1     RESW    1
```

```
N2     RESW    1
-      END     -
```

Output of pass1.c

Namtab.txt

```
EX1
```

Deftab.txt

```
EX1  &A,&B
LDA  &A
STA  &B
MEND
```

Pass2.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void main()
{
    FILE *f1, *f2, *f3, *f4, *f5;
    int i, len;
    char mne[20], opnd[20], la[20], name[20], mne1[20], opnd1[20], arg[20];
    f1 = fopen("input.txt", "r");
    f2 = fopen("namtab.txt", "r");
    f3 = fopen("deftab.txt", "r");
    f4 = fopen("argtab.txt", "w+");
    f5 = fopen("output.txt", "w");
    fscanf(f1, "%s%s%s", la, mne, opnd);
    while (strcmp(mne, "END") != 0)
    {
        if (strcmp(mne, "MACRO") == 0)
        {
            fscanf(f1, "%s%s%s", la, mne, opnd);
            while (strcmp(mne, "MEND") != 0)
                fscanf(f1, "%s%s%s", la, mne, opnd);
        }
        else
        {
            fscanf(f2, "%s", name);
            if (strcmp(mne, name) == 0)
            {
                len = strlen(opnd);
                for (i = 0; i < len; i++)
                {
                    if (opnd[i] != ',')
                        fprintf(f4, "%c", opnd[i]);
                    else
                        fprintf(f4, "\n");
                }
                fseek(f2, SEEK_SET, 0);
                fseek(f4, SEEK_SET, 0);
                fscanf(f3, "%s%s", mne1, opnd1);
                fprintf(f5, ".\t%s\t%s\n", mne1, opnd);
                fscanf(f3, "%s%s", mne1, opnd1);
                while (strcmp(mne1, "MEND") != 0)
                {
                    if ((opnd1[0] == '&'))
                    {
                        fscanf(f4, "%s", arg);
                        fprintf(f5, "-\t%s\t%s\n", mne1, arg);
                    }
                    else
                        fprintf(f5, "-\t%s\t%s\n", mne1, opnd1);
                    fscanf(f3, "%s%s", mne1, opnd1);
                }
            }
            else
                fprintf(f5, "%s\t%s\t%s\n", la, mne, opnd);
        }
        fscanf(f1, "%s%s%s", la, mne, opnd);
    }
    fprintf(f5, "%s\t%s\t%s\n", la, mne, opnd);
    fclose(f1);
    fclose(f2);
    fclose(f3);
    fclose(f4);
    fclose(f5);
    printf("\nPass 2 of 2 Pass Macroprocessor is Successful.");
}
```

Argtab.txt

```
N1
N2
```

Output.txt

```
SAMPLE   START      1000
.       EX1  N1,N2
-       LDA  N1
-       STA  N2
N1    RESW      1
N2    RESW      1
-       END -
```

Implement Single Pass Macro Processor

Onepass.c

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
void main()
{
   FILE *f1, *f2, *f3, *f4, *f5;
   int len, i, pos = 1;
   char arg[20], mne[20], opnd[20], la[20], name[20], mne1[20], opnd1[20], pos1[10], pos2[10];
   //clrscr();
   f1 = fopen("input.txt", "r");
   f2 = fopen("namtab.txt", "w+");
   f3 = fopen("deftab.txt", "w+");
   f4 = fopen("argtab.txt", "w+");
   f5 = fopen("op.txt", "w+");
   fscanf(f1, "%s%s%s", la, mne, opnd);
   while (strcmp(mne, "END") != 0)
   {
      if (strcmp(mne, "MACRO") == 0)
      {
         fprintf(f2, "%s\n", la);
         fseek(f2, SEEK_SET, 0);
         fprintf(f3, "%s\t%s\n", la, opnd);
         fscanf(f1, "%s%s%s", la, mne, opnd);
         while (strcmp(mne, "MEND") != 0)
         {
            if (opnd[0] == '&')
            {
               itoa(pos, pos1, 5);
               strcpy(pos2, "?");
               strcpy(opnd, strcat(pos2, pos1));
               pos = pos + 1;
            }
            fprintf(f3, "%s\t%s\n", mne, opnd);
            fscanf(f1, "%s%s%s", la, mne, opnd);
         }
         fprintf(f3, "%s", mne);
```

```
    }
    else
    {
        fscanf(f2, "%s", name);
        if (strcmp(mne, name) == 0)
        {
            len = strlen(opnd);
            for (i = 0; i < len; i++)
            {
                if (opnd[i] != ',')
                    fprintf(f4, "%c", opnd[i]);
                else
                    fprintf(f4, "\n");
            }
            fseek(f3, SEEK_SET, 0);
            fseek(f4, SEEK_SET, 0);
            fscanf(f3, "%s%s", mne1, opnd1);
            fprintf(f5, ".\t%s\t%s\n", mne1, opnd);
            fscanf(f3, "%s%s", mne1, opnd1);
            while (strcmp(mne1, "MEND") != 0)
            {
                if ((opnd[0] == '?'))
                {
                    fscanf(f4, "%s", arg);
                    fprintf(f5, "-\t%s\t%s\n", mne1, arg);
                }
                else
                    fprintf(f5, "-\t%s\t%s\n", mne1, opnd1);
                fscanf(f3, "%s%s", mne1, opnd1);
            }
        }
        else
            fprintf(f5, "%s\t%s\t%s\n", la, mne, opnd);
    }
    fscanf(f1, "%s%s%s", la, mne, opnd);
}
fprintf(f5, "%s\t%s\t%s", la, mne, opnd);
fclose(f1);
fclose(f2);
fclose(f3);
fclose(f4);
```

```
    fclose(f5);
    printf("files to be viewed \n");
    printf("1. argtab.txt\n");
    printf("2. namtab.txt\n");
    printf("3. deftab.txt\n");
    printf("4. op.txt\n");
    getch();
}
```

Input.txt

```
EX1     MACRO   &A,&B
-       LDA     &A
-       STA     &B
-       MEND    -
SAMPLE  START   1000
-       EX1     N1,N2
N1      RESW    1
N2      RESW    1
-       END     -
```

Namtab.txt

```
EX1
```

Deftab.txt

```
EX1  &A,&B
LDA  ?1
STA  ?2
MEND
```

Argtab.txt

```
N1
N2
```

Op.txt

```
SAMPLE   START     1000
.       EX1  N1,N2
-       LDA  ?1
-       STA  ?2
N1      RESW      1
N2      RESW      1
-       END -
```

Implement Pass2 of Two Pass Assembler

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
   int length, start;
   char operand [20], label [20],opcode [20], mn[20],code [20],locctr [20], loc
[20],symbol[20],hexvalue [20];
   FILE *inter,*optab, *symtab, *objectcode, *len;

   inter=fopen("INTER.txt","r");
   optab=fopen("OPTAB.txt","r");
   symtab=fopen("SYMTAB.txt","r");
   len=fopen("LEN.txt","r");
   fscanf(len,"%s",loc);
   printf("%s\n",loc);
   objectcode=fopen("OC.txt","w");
   fscanf(inter,"%s\t%s\t%s\n",locctr, label, opcode);
   printf("\n BEFOER IF
LOCCTR=%s\tLaBEL=%s\tOPCODE=%s",locctr,label,opcode);
   if (strcmp(label,"START")==0)
   {
      printf("\nH^%06s^%06s",opcode,loc);
      fprintf(objectcode,"H^%06s^%06s",opcode,loc);
   }
   int c=0;
   fscanf(inter,"%s\t%s\t%s\t%s",locctr,label,opcode,operand);
   fprintf(objectcode,"\nT^00%s^000%s",locctr,loc);
   printf("\n BEFORE
WILELOCCTR=%s\tLaBEL=%s\tOPCODE=%s\toPERAND=%s",locctr,label,o
pcode,operand);
   while(!feof(inter))
   {
  printf("\nLOCCTR=%s\tLaBEL=%s\tOPCODE=%s\toPERAND=%s",locctr,lab
el,opcode,operand);
      if (strcmp(label,"**")==0)
      {
         printf("\nInsid if");
         fscanf(optab,"%s\t%s",code,mn);
```

```
            while(!feof(optab))
            {
                printf("\nOPCODE  %s    MN    %s",code,mn);
                if (strcmp(opcode,code)==0)
                {
                    rewind(optab);
                    break;
                }
                fscanf(optab,"%s\t%s",code,mn);
            }
            fscanf(symtab,"%s\t%s",symbol,hexvalue);
            printf("\nOPCODE final %s    MN    %s",code,mn);

            while (!feof(symtab))
            {   printf("\nSYMBOL  %s    HEX    %s",symbol,hexvalue);
                if(strcmp(operand,symbol)==0)
                {
                    rewind(symtab);
                    break;
                }
                fscanf(symtab,"%s\t%s",symbol,hexvalue);
            }
            fprintf(objectcode,"^%s%s",mn,hexvalue);
            printf("###%s  %s",mn,hexvalue);
            //fscanf(inter,"%s\t%s\t%s\t%s",locctr,label,opcode,operand);

printf("\nLOCCTR=%s\tLaBEL=%s\tOPCODE=%s\toPERAND=%s",locctr,label
,opcode,operand);
        }
        else
        {
            printf("INSIDE ELSE");
            if (strcmp(opcode,"RESB")==0||strcmp(opcode,"RESW")==0)
            {
                fscanf(inter,"%s\t%s\t%s\t%s",locctr,label,opcode,operand);

printf("\nLOCCTR=%s\tLaBEL=%s\tOPCODE=%s\toPERAND=%s",locctr,label
,opcode,operand);
                continue;
            }
            else if (strcmp(opcode,"WORD")==0)
            {
                printf("\nWORD %06d",atoi(operand));
                fprintf(objectcode,"^%06d",atoi(operand));
            }
            else if (strcmp(opcode,"BYTE")==0)
            {
                int i=0;
                if (operand[0]=='X')
                {
                    for (int j=2;j<strlen(operand)-1;j++)
                    {
                        hexvalue[i]=operand[j];
                        i++;
                    }
                    fprintf(objectcode,"^%06s",hexvalue);
                }
                else if (operand[0]=='C')
                {
                    for (int j=2;j<strlen(operand)-1;j++)
                    {
                        hexvalue[i]=operand[j];
                        i++;
                    }
                    fprintf(objectcode,"^%06x",hexvalue);
                }
            }
        }
        fscanf(inter,"%s\t%s\t%s\t%s",locctr,label,opcode,operand);
        printf("\nlabel %s  count  %d",label,c);
    }
    fprintf(objectcode,"\nE^%06s",loc);
    fclose(objectcode);
    fclose(len);
    fclose(inter);
    fclose(optab);
}
```

INTER.txt

```
MUL START     1000
1000 **     LDA  ALPHA
1003 **     MUL BETA
```

```
1006 **      STA  GAMMA
1009 ALPHA        WORD     2
1012 BETA         WORD     4
1015 GAMMA        RESW     1
**      END **
```

OPTAB.txt

```
LDA  00
STA  0C
ADD  18
SUB  1C
MUL  20
DIV  24
```

OC.txt

```
H^001000^000018
T^001000^00018^001009^201012^0C1015^000002^000004
E^000018
```