System Programming lab work (Credits: Fayaz Lone)

Write a program in C/C++ to simulate the working of a single pass macro-processor capable of handling

```
#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 FILES

Start here | main.c | Input.txt | Deftab.txt

```
1   EX1 MACRO &A,&B
2   - LDA &A
3   - STA &B
4   - MEND -
5   SAMPLE START 1000
6   - EX1 N1,N2
7   N1 RESW 1
8   N2 RESW 1
9   - END -
10
```

main.c | Input.txt | Deftab.txt

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

main.c | Input.txt | Deftab.txt | Argtab.txt

```
N1
N2
```

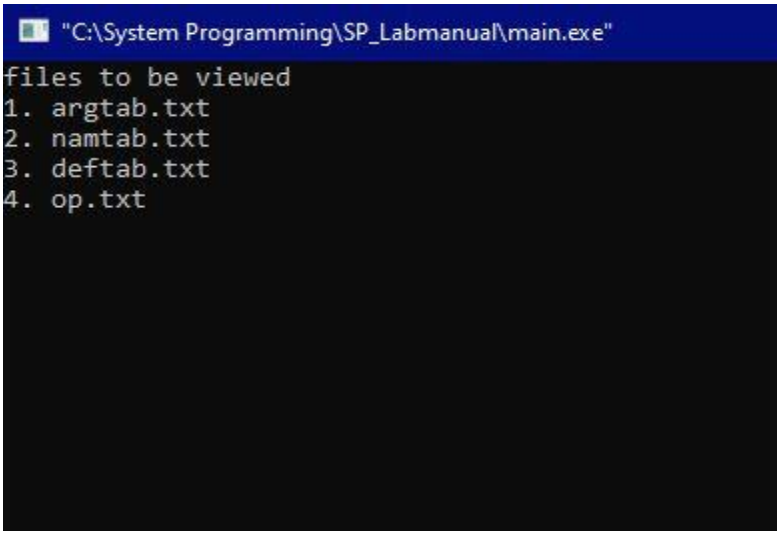
main.c | Input.txt | Deftab.txt | Argtab.txt | Namtab.txt

```
EX1
```

main.c | Input.txt | Deftab.txt | Argtab.txt | Namtab.txt | Op.txt

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

OUTPUT



Display symbol table of a sample program stored in file named cpp.txt.The symbol table is stored in the form of text file named symtab.txt

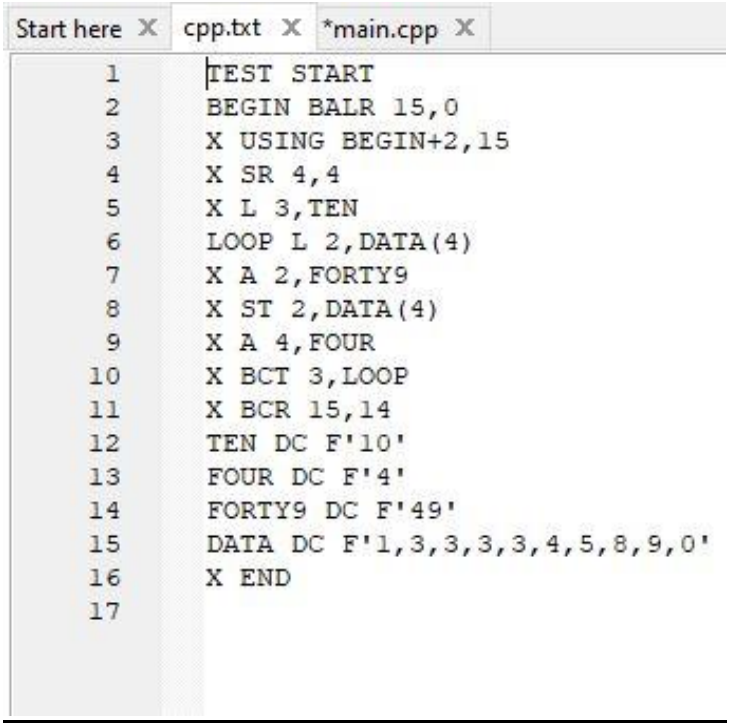
```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<iostream>
#include<string.h>
using namespace std;
FILE *f1,*f3;//f1 is pointer for spp.txt,f3 is for symbol table
int lc,sa,i=0,j=0,k,len1=1; //lc is location counter,len1 keeps track of instr length
char name[20],opnd[30],la[20],mne[20],s1[20];
//these hold various parts of inst
void addsymbol()
{ //if(strcmp(la,"BEGIN")==0) //for first instru
```

```
if(strcmp(mne,"BALR")==0) //if mnemonic is balr length of instruction is 2
len1=2;
else if(strcmp(mne,"START")==0)
len1=1;
else
len1=4;
if(strcmp(la,"X")!=0 && strcmp(mne,"EQU")!=0) //if label is present in the instruction
//and its not a constant then its relative
fprintf(f3,"%s\t\t\t%d\t\t\t%s\n",la,len1,"R");
else if(strcmp(la,"X")!=0 && strcmp(mne,"EQU")==0) //if label is present in the instruction
//and its a constant then its absolute
fprintf(f3,"%s\t\t\t%d\t\t\t%s\n",la,len1,"A");
}
void readf(FILE *fp)
{
rewind(fp);
char ch=fgetc(fp);
while(ch!=EOF)
{
if(ch!='X')
cout<<ch; ch=fgetc(fp);
}
getch();
```

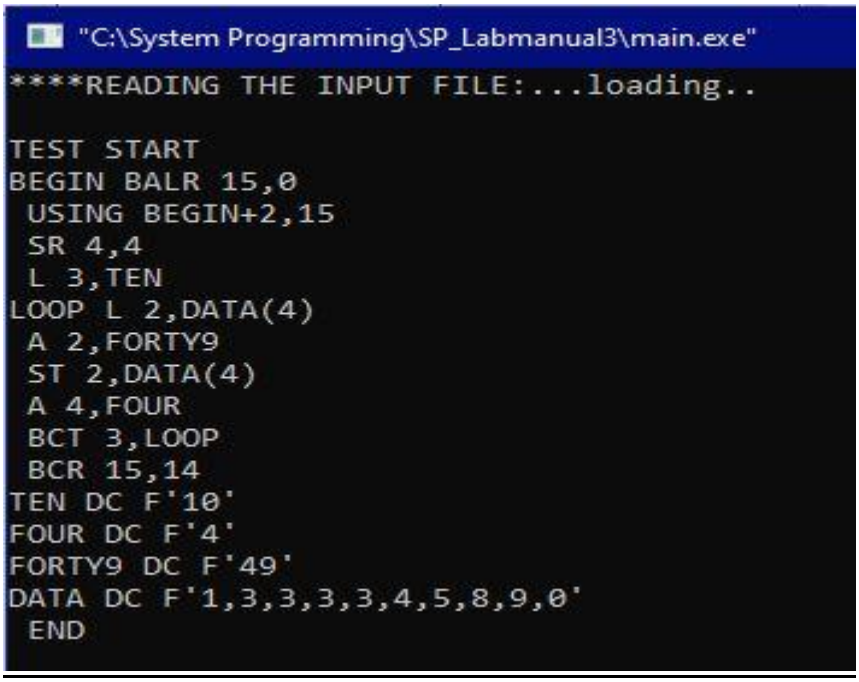
```
}
int main()
{
//clrscr();
f1=fopen("cpp.txt","r"); //open the text file contaning program in read mode
// f2=fopen("optab.txt","r");
f3=fopen("symtab.txt","w+"); //table files to be created are opened in write mode
char ch;
fscanf(f1,"%s%s",la,mne); //takes ist instr,stores its label in la, opcode mnemonic in mne
sa=atoi(opnd); //converts string to int and stores value in sa
strcpy(name,la); //copies label to name
lc=sa;//sets location counter to 0 when ist instruction is encountered ,sets location counter
fprintf(f3,"%s\t\t\t%s\t\t\t%s\n\n","symbol","length","Relocation");
fprintf(f3,"%s\t\t\t%s\t\t\t%s\n",la,"1","R"); //add ist label to symbol table
//********now dealing with next instructions
//while(strcmp(mne,"END")!=0)
for(j=0;j<15;j++)
{
fscanf(f1,"%s%s%s",la,mne,opnd);
addsymbol();//add label and length of instruction to symbol table
lc+=1; //move to location counter+1
}//endwhile
//READ FILES ONE BY ONE 2
```

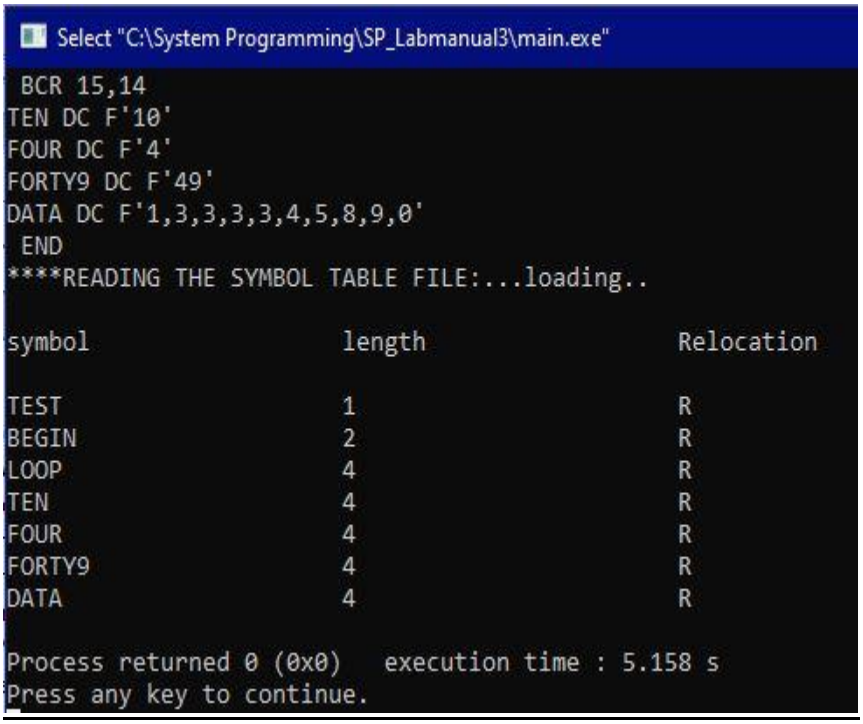
```
cout<<"****READING THE INPUT FILE:...loading.."<<endl<<endl;
readf(f1); //display original program
fclose(f3);
f3=fopen("symtab.txt","r");
getch();
//clrscr();
cout<<"****READING THE SYMBOL TABLE FILE:...loading.."<<endl<<endl;
readf(f3); //display symbol table
getch();
fclose(f3);
getch();
}
```

INPUT FILE



OUTPUT


"C:\System Programming\SP_Labmanual3\main.exe"
****READING THE INPUT FILE:...loading..
TEST START
BEGIN BALR 15,0
USING BEGIN+2,15
SR 4,4
L 3,TEN
LOOP L 2,DATA(4)
A 2,FORTY9
ST 2,DATA(4)
A 4,FOUR
BCT 3,LOOP
BCR 15,14
TEN DC F'10'
FOUR DC F'4'
FORTY9 DC F'49'
DATA DC F'1,3,3,3,3,4,5,8,9,0'
END


Select "C:\System Programming\SP_Labmanual3\main.exe"
BCR 15,14
TEN DC F'10'
FOUR DC F'4'
FORTY9 DC F'49'
DATA DC F'1,3,3,3,3,4,5,8,9,0'
END
****READING THE SYMBOL TABLE FILE:...loading..
symbol length Relocation
TEST 1 R
BEGIN 2 R
LOOP 4 R
TEN 4 R
FOUR 4 R
FORTY9 4 R
DATA 4 R
Process returned 0 (0x0) execution time : 5.158 s
Press any key to continue.

Display symbol table and psuedo op table for a given program stored in a text file named spp.txt.The symbol table is stored in the from of text file named symtab.txt, psuedo op table is stored in pot.txt and machine op table is stored as mot.txt

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<iostream>
using namespace std;
#include<string.h>
FILE *f1,*f2,*f3,*f4,*f5;//f1 is pointer for spp.txt,f2 is for mot, f3 is for symbol table
//f4 is for psuedo op table
int lc,sa,i=0,j=0,k,len1=1; //lc is location counter,len1 keeps track of instr length
char name[20],opnd[30],la[20],mne[20],s1[20];
//these hold various parts of inst
void addsymbol()
{ //if(strcmp(la,"BEGIN")==0) //for first instru
if(strcmp(mne,"BALR")==0) //if mnemonic is balr length of instruction is 2
len1=2;
else if(strcmp(mne,"START")==0)
len1=1;
else
len1=4;
```

```
if(strcmp(la,"X")!=0 && strcmp(mne,"EQU")!=0) //if label is present in the instruction

//and its not a constant then its relative

fprintf(f3,"%s\t\t\t%d\t\t\t%s\n",la,len1,"R");

else if(strcmp(la,"X")!=0 && strcmp(mne,"EQU")==0) //if label is present in the instruction

//and its a constant then its absolute

fprintf(f3,"%s\t\t\t%d\t\t\t%s\n",la,len1,"A");

}

void addpot()

{

fprintf(f4,"%s\t\t\t\t\t\tP1%s\n",mne,mne);

}

void addmot()

{
if(strcmp(mne,"A")==0||strcmp(mne,"L")==0||strcmp(mne,"BCT")==0||strcmp(mne,"L")==0)

fprintf(f2,"%s\t\t\t\t%s\t\t\t\t%s\n",mne,"4","001"); //FOR RX FORMAT

else if(strcmp(mne,"SR")==0||strcmp(mne,"BR")==0)

fprintf(f2,"%s\t\t\t\t%s\t\t\t\t%s\n",mne,"2","000"); //FOR RR FROMAT

}

void readf(FILE *fp)

{

rewind(fp);

char ch=fgetc(fp);

while(ch!=EOF)
```

```
{
if(ch!='X')
cout<<ch; ch=fgetc(fp);
}
getch();
}
int main()
{
//clrscr();
f1=fopen("spp.txt","r"); //open the text file contaning program in read mode
// f2=fopen("optab.txt","r");
f3=fopen("symtab.txt","w+"); //table files to be created are opened in write mode
f2=fopen("mot.txt","w+");
f4=fopen("pot.txt","w+");
char ch;
fscanf(f1,"%s%s",la,mne); //takes ist instr,stores its label in la, opcode mnemonic in mne
sa=atoi(opnd); //converts string to int and stores value in sa
strcpy(name,la); //copies label to name
lc=sa;//sets location counter to 0 when ist instruction is encountered ,sets location counter
fprintf(f3,"%s\t\t\t%s\t\t\t%s\n\n","symbol","length","Relocation");
fprintf(f3,"%s\t\t\t%s\t\t\t%s\n",la,"1","R"); //add ist label to symbol table
fprintf(f4,"%s\t\t\t\t\t%s\n\n","PSEUDO-OP","PROCEDURE");
```

```
fprintf(f2,"%s\t\t%s\t\t%s\n\n","MACHINE OP","INSTRUCTION
LENGTH","INSTRUCTION FORMAT");

//********now dealing with next instructions

//while(strcmp(mne,"END")!=0)

for(j=0;j<15;j++)

{

fscanf(f1,"%s%s%s",la,mne,opnd);

addsymbol();//add label and length of instruction to symbol table

if(strcmp(mne,"START")==0||strcmp(mne,"END")==0||strcmp(mne,"USING")==0
||strcmp(mne,"DC")==0||strcmp(mne,"EQU")==0)

addpot(); //add psuedo ops to pot.txt

else //if not a psuedo op add to machine op table

addmot();

lc+=1; //move to location counter+1

// cout<<"la "<<la<<" mne "<<mne<<" opnd "<<opnd;

// cout<<" lc"<<lc<<endl;

}//endwhile

//READ FILES ONE BY ONE 2

cout<<"****READING THE INPUT FILE:...loading.."<<endl<<endl;

readf(f1); //display original program

fclose(f3);

f3=fopen("symtab.txt","r");

getch();

//clrscr();

cout<<"****READING THE SYMBOL TABLE FILE:...loading.."<<endl<<endl;
```

```
readf(f3); //display symbol table
getch();
fclose(f3);
//clrscr();
cout<<"****READING THE PSUEDO-OP TABLE FILE:...loading.."<<endl<<endl;
fclose(f4);
f4=fopen("pot.txt","r");
readf(f4); //display psuedo op table
fclose(f4);
fclose(f2);
//clrscr();
cout<<"****READING THE MACHINE-OP TABLE FILE:...loading.."<<endl<<endl;
f2=fopen("mot.txt","r");
readf(f2);
getch();
fclose(f2);
getch();
}
```

INPUT FILE

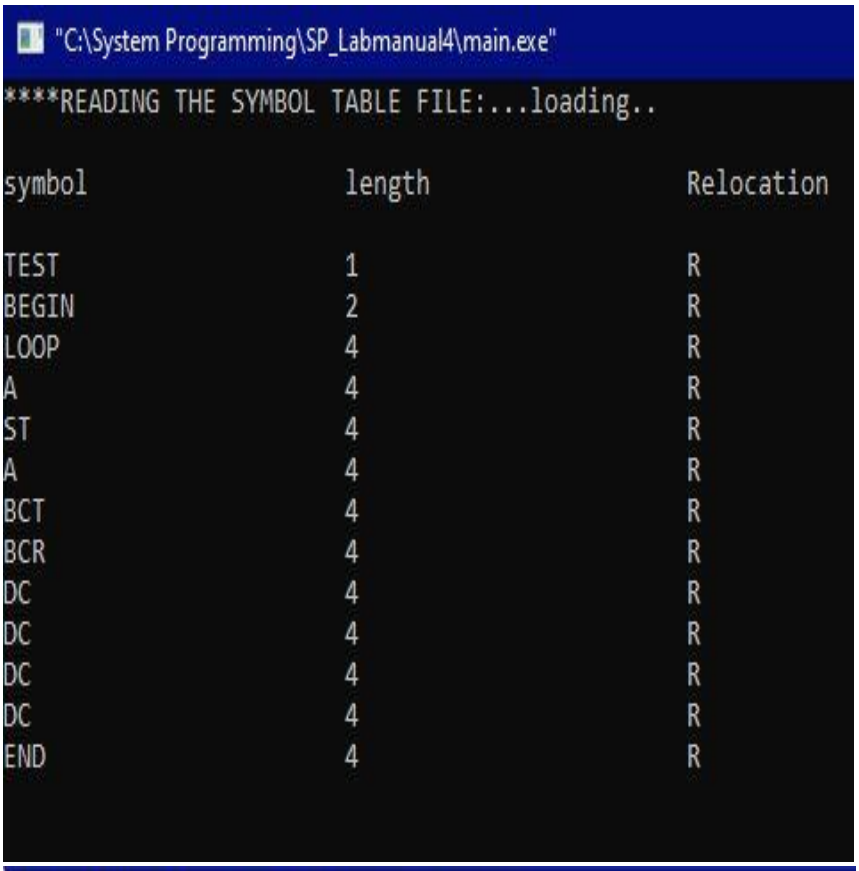
Start here | *main.cpp | spp.txt

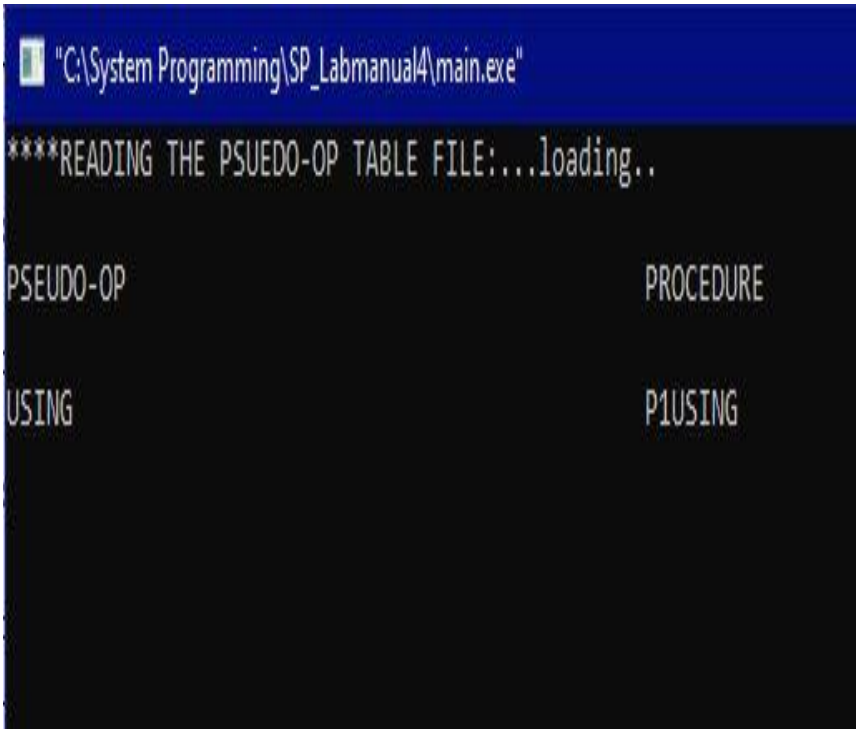
```
1     TEST START
2     BEGIN BALR 15,0
3     X USING BEGIN+2,15
4     X SR 4,4
5     X L 3,TEN
6     LOOP L 2,DATA(4)
7     A 2,FORTY9
8     X ST 2,DATA(4)
9     X A 4,FOUR
10    X BCT 3,LOOP
11    X BCR 15,14
12    TEN DC F'10'
13    FOUR DC F'4'
14    FORTY9 DC F'49'
15    DATA DC F'1,3,3,3,3,4,5,8,9,0'
16    X END
17
```

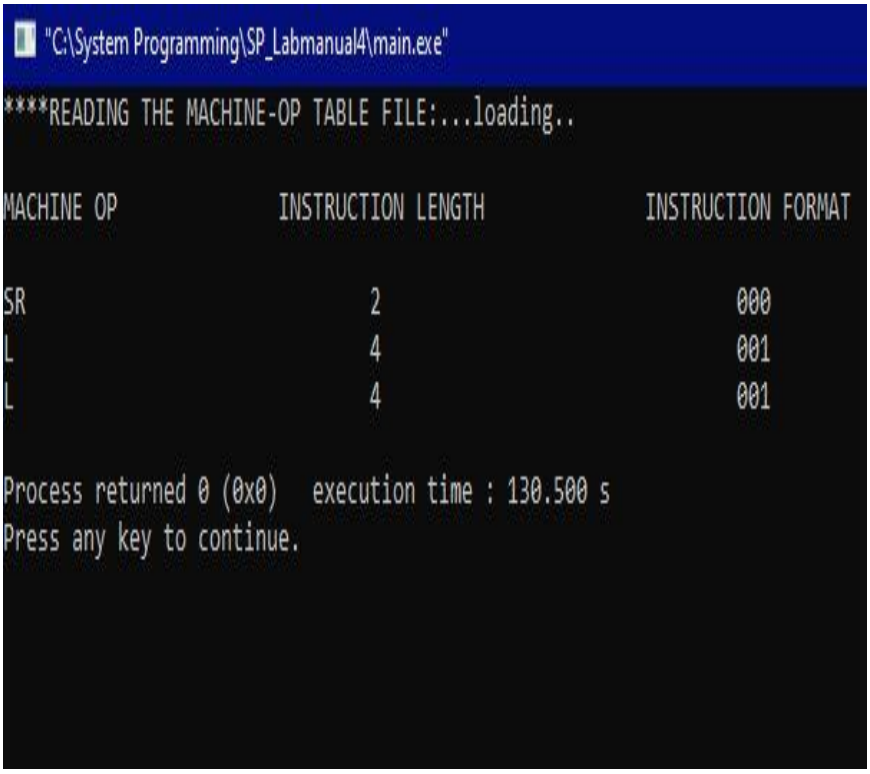
OUTPUT

```
"C:\System Programming\SP_Labmanual4\main.exe"
****READING THE INPUT FILE:...loading..

TEST START
BEGIN BALR 15,0
 USING BEGIN+2,15
 SR 4,4
 L 3,TEN
LOOP L 2,DATA(4)
A 2,FORTY9
 ST 2,DATA(4)
 A 4,FOUR
 BCT 3,LOOP
 BCR 15,14
TEN DC F'10'
FOUR DC F'4'
FORTY9 DC F'49'
DATA DC F'1,3,3,3,3,4,5,8,9,0'
 END
```


"C:\System Programming\SP_Labmanual4\main.exe"
****READING THE SYMBOL TABLE FILE:...loading..
symbol length Relocation
TEST 1 R
BEGIN 2 R
LOOP 4 R
A 4 R
ST 4 R
A 4 R
BCT 4 R
BCR 4 R
DC 4 R
DC 4 R
DC 4 R
DC 4 R
END 4 R


"C:\System Programming\SP_Labmanual4\main.exe"
****READING THE PSUEDO-OP TABLE FILE:...loading..
PSEUDO-OP PROCEDURE
USING P1USING


"C:\System Programming\SP_Labmanual4\main.exe"
****READING THE MACHINE-OP TABLE FILE:...loading..
MACHINE OP INSTRUCTION LENGTH INSTRUCTION FORMAT
SR 2 000
L 4 001
L 4 001
Process returned 0 (0x0) execution time : 130.500 s
Press any key to continue.

"C" program for the implementation of pass one of a two pass assembler

```
//Pass -1 of 2-pass assembler
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
FILE *f1,*f2,*f3,*f4;
int lc,sa,l,op1,o,len;
char m1[20],la[20],op[20],otp[20];
//clrscr();
f1=fopen("input.txt","r");
f3=fopen("symtab.txt","w");
fscanf(f1,"%s %s %d",la,m1,&op1);
if(strcmp(m1,"START")==0)
{
 sa=op1;
 lc=sa;
 printf("\t%s\t%s\t%d\n",la,m1,op1);
 }
 else
 lc=0;
fscanf(f1,"%s %s",la,m1);
```

```
while(!feof(f1))
{
 fscanf(f1,"%s",op);
 printf("\n%d\t%s\t%s\t%s\n",lc,la,m1,op);
 if(strcmp(la,"-")!=0)
 {
 fprintf(f3,"\n%d\t%s\n",lc,la);
 }
 f2=fopen("optab.txt","r");
 fscanf(f2,"%s %d",otp,&o);
 while(!feof(f2))
 {
  if(strcmp(m1,otp)==0)
  {
    lc=lc+3;
    break;
  }
  fscanf(f2,"%s %d",otp,&o);
  }
  fclose(f2);
  if(strcmp(m1,"WORD")==0)

    {
   lc=lc+3;
```

```
}
else if(strcmp(m1,"RESW")==0)
{
 op1=atoi(op);
 lc=lc+(3*op1);
 }
 else if(strcmp(m1,"BYTE")==0)
 {
 if(op[0]=='X')
   lc=lc+1;
   else
   {
   len=strlen(op)-2;
   lc=lc+len;}
 }
 else if(strcmp(m1,"RESB")==0)
 {
  op1=atoi(op);
  lc=lc+op1;
  }
 fscanf(f1,"%s%s",la,m1);
 }
 if(strcmp(m1,"END")==0)
 {
```

```
printf("Program length =\n%d",lc-sa);
}
fclose(f1);
fclose(f3);
getch();
}
```

INPUT FILES

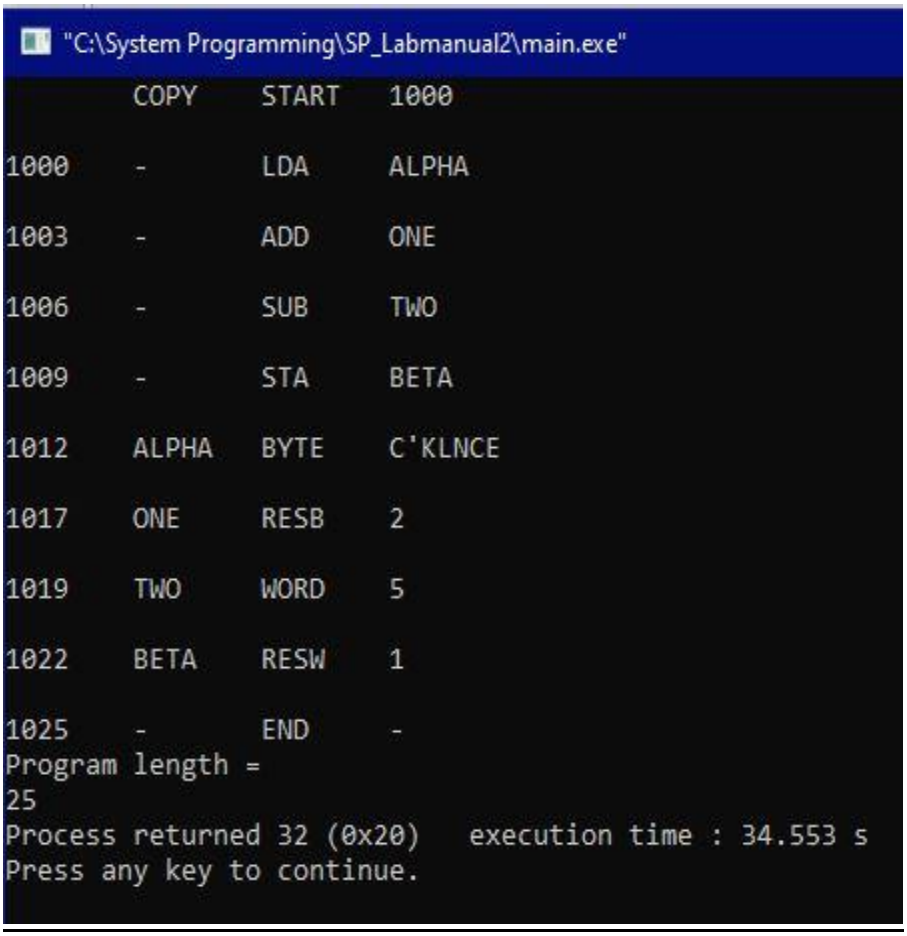
Start here | main.c | Optab.txt | input.txt

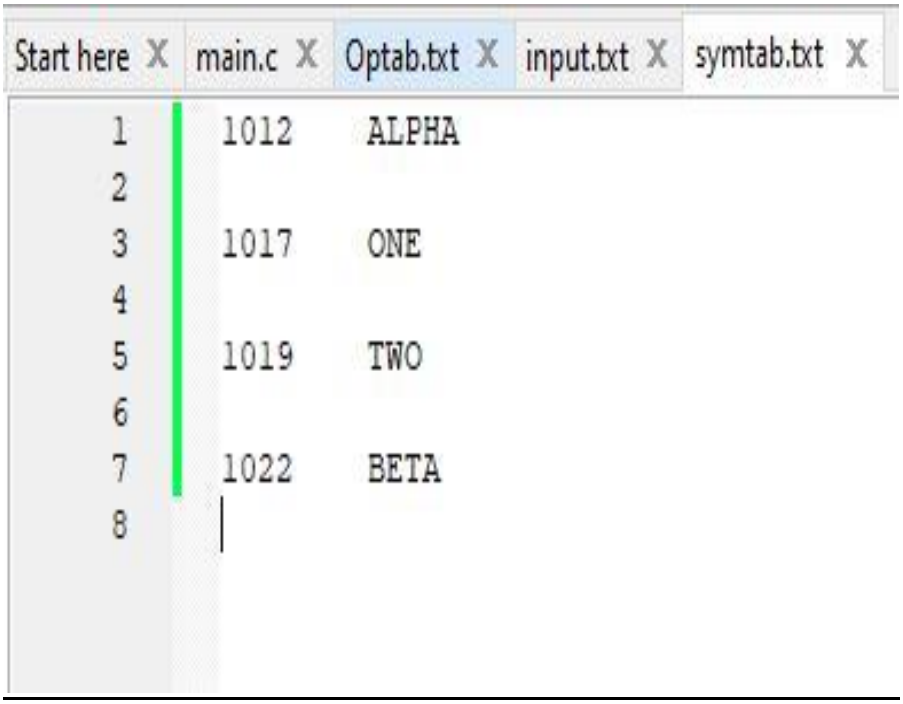
```
1     COPY    START   1000
2     -   LDA ALPHA
3     -   ADD ONE
4     -   SUB TWO
5     -   STA BETA
6     ALPHA   BYTE    C'KLNCE
7     ONE RESB    2
8     TWO WORD    5
9     BETA    RESW    1
10    -   END -
11
12
```

Start here | main.c | Optab.txt | input.txt

```
1     LDA 00
2     STA 23
3     ADD 01
4     SUB 05
5
```

OUTPUT


"C:\System Programming\SP_Labmanual2\main.exe"
COPY START 1000
1000 - LDA ALPHA
1003 - ADD ONE
1006 - SUB TWO
1009 - STA BETA
1012 ALPHA BYTE C'KLNCE
1017 ONE RESB 2
1019 TWO WORD 5
1022 BETA RESW 1
1025 - END -
Program length =
25
Process returned 32 (0x20) execution time : 34.553 s
Press any key to continue.


Start here
main.c
Optab.txt
input.txt
symtab.txt
1 1012 ALPHA
2
3 1017 ONE
4
5 1019 TWO
6
7 1022 BETA
8

//Write a program in C/C++ to simulate the working of an absolute loader?

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
  FILE *fp;
  int i,addr1,l,j,staddr1;
  char name[10],line[50],name1[10],addr[10],rec[10],ch,staddr[10];
//  clrscr();
  printf("enter program name:" );
  scanf("%s",name);
  fp=fopen("abssrc.txt","r");
  fscanf(fp,"%s",line);
  for(i=2,j=0;i<8,j<6;i++,j++)
    name1[j]=line[i];
    name1[j]='\0';
  printf("name from obj. %s\n",name1);
  if(strcmp(name,name1)==0)
   {
    do
    {
```

```
fscanf(fp,"%s",line);
if(line[0]=='T')
{
for(i=2,j=0;i<8,j<6;i++,j++)
staddr[j]=line[i];
staddr[j]='\0';
staddr1=atoi(staddr);
i=12;
while(line[i]!='$')
{
  if(line[i]!='^')
  {
    printf("00%d \t %c%c\n", staddr1,line[i],line[i+1]);
    staddr1++;
    i=i+2;
  }
  else i++;
}
}
else if(line[0]='E')
fclose(fp);
}while(!feof(fp));
}
```

```
getch();
}
```

INPUT FILE

Start here | main.c | ABSSRC.TXT

```
1  H^SAMPLE^001000^0035
2  T^001000^0C^001003^071009$
3  T^002000^03^111111$
4  E^001000
5
```

OUTPUT

```
"C:\System Programming\Question7_SP_Labmanual\main.exe"
enter program name:
SAMPLE
name from obj. SAMPLE
001000   00
001001   10
001002   03
001003   07
001004   10
001005   09
002000   11
002001   11
002002   11
```

Write a program in C/C++ to simulate the working of a direct-linking loader?

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

void main()
{
  int i,j,k,pgmaddr,csaddr,cslth=0,z,addr1,l1;
  char line[50],name[10],len[10],la[10],addr[10];
  FILE *fp1;
//  clrscr();
  printf("Enter the starting address\n");
  scanf("%d",&pgmaddr);
  csaddr=pgmaddr;
  for(k=0;k<2;k++)
  {
    if(k==0)
     fp1=fopen("linkin.txt","r");
if(k==1)
     fp1=fopen("linkin1.txt","r");
do
{
 fscanf(fp1,"%s",line);
if(line[0]=='H')
{
```

```
  for(i=2,j=0;i<8,j<6;i++,j++)
  name[j]=line[i];
name[j]='\0';
for(i=16,j=0;i<20,j<5;i++,j++)
 len[j]=line[i];
len[j]='\0';
cslth=atoi(len);
printf("%s\t\t%d\t%s\n",name,csaddr,len);
}
else if(line[0]=='D')
{
  i=2;
  j=0;
do
{
do
{
la[j++]=line[i++];
}
while(line[i]!='^');
la[j]='\0';
j=0;i++;
do
{
```

```
addr[j++]=line[i++];
}
while(line[i]!='^');
i++;
addr[j]='\0';
addr1=atoi(addr)+csaddr;
j=0;
printf("%s\t\t%d\n",la,addr1);
} while(line[i]!='\0');
}
else if(line[0]=='R'||'T')
z=0;
else if(line[0]=='E')
fclose(fp1);
}while(!feof(fp1));
csaddr=csaddr+cslth;
}
getch();
}
```

<u>INPUT FILES</u>

Start here | main.c | linkin.txt | linkin1.txt

```
1  H^PROGA1^000000^0073
2  D^LISTA^000024^ENDA^000027^
3  R^LISTB
4  T^000000^06^000024^010027
5  E^000000
6
```

Start here | main.c | linkin.txt | linkin1.txt

```
1  H^PROGA2^000000^0089
2  D^LISTB^000047^
3  E
4
```

OUTPUT

```
"C:\System Programming\Question8_SP_Labmanual\main.exe"
Enter the starting address
2000
PROGA1          2000    0073
LISTA           2024
ENDA            2027
PROGA2          2073    0089
LISTB           2120
```