SIC/XE Assembler

Phase 2

Requirements specification

1. To execute pass1 you should enter pass1<source-file-name>.
2. To execute pass2 you should enter pass2<source-file-name>.
3. To execute the whole assembly, you should enter assemble<source-file-name>.
4. The source file for the main program is named assemble.cpp.
5. Handling phase 2 errors through a parser.
6. Generating the object code for each line.
7. Generating the object program at the end.
8. The output of the assembler should include (at least):
   a) Object-code file.
   b) A report at the end of pass2. Pass1 and Pass2 errors should be included as part of the assembler report, exhibiting both the erroneous lines of source code and the error.
9. The assembler should support:

a) EQU and ORG statements.

b) Simple expression evaluation. A simple expression includes simple (A <op> B) operand arithmetic, where <op> is one of +, -, *, / and no spaces surround the operation.

Design

- We used Object Oriented Programming Concepts.
- Classes from phase 1 are included in a source folder named passOne and they are:
    1. Parsing class: in which the regex handles all entered lines and makes sure that there are no syntax errors.
    2. addressList class: which sets the address for each line of code and produces the symbol table.
    3. Output1 class: which prints the output of pass1 only.
- Classes from phase 2 are included in a source folder named passTwo.
    1. oCode class: which generates the object code for each line.
    2. pProgram class: which generates the object program.
    3. Output2 class: which prints the output of pass2 only.
- A final class called output3 which prints the output of the whole assembler.
- The main of assemble.cpp which chooses which output to display based on what the user enters.

Main Data structures

Maps and vectors

- We store all operation names in a map whose key is the name of the operation and it has a pair of values; the first is the length of the operation in bits and the second is the code of the operation in hexadecimal.
- We use maps to store the symbol table.
- We use vectors to store all labels, operations, operands, addresses and object codes.
- We use vectors to store all lines of code.
- We use vectors to store the indices of all lines that have errors.

Maps and vectors

- We use vector to store all possible instructions to handle them as :

  a) 2-byte instructions with 1 or 2 symbolic register reference.

  b) 3-byte PC-relative with symbolic operand to include immediate, indirect, and indexed addressing

  c) 3-byte absolute with non-symbolic operand to include immediate, indirect, and indexed addressing

  d) 4-byte absolute with symbolic or non-symbolic operand to include immediate, indirect, and indexed addressing 5

Algorithms Description

Parsing class

-We create a group of regex where each regex handles a type of instruction.

-By looping on the file and check that each line is acceptable by a regex .

-The line may not be acceptable if the instruction is unsupported instructions or unsupported operand for the given instruction.

-If the line is not acceptable by any regex, then the output print "Syntax error".

-If the last line is not END , it will print "the last line is not END ".

AddressList Class

-if the given mnemonic in the problem contains in the map we add to the address the value in stored in pair->first

-if mnemonic in the problem stored in the vector if its BYTE we increase the address by number of bytes stored in given problem ex C'AB' so increase by 2,

- if it is RESB we increase address by given number of bytes

-if it is RESW we increase by given number of words multiplyed by 3

-if it is WORD we increase address by 3

Output classes

- In case of wanting to print pass1 only or pass2 only:
    1. They take the vectors that contain the Mnemonic Op-codes, operands, addresses, labels, object codes, the map that contains the symbol table and print them with the line number.
    2. Print all existing errors.
- In case of wanting to print the whole assembler:
    1. They take the vectors that contain the Mnemonic Op-codes, operands, addresses, labels, object codes, the map that contains the symbol table and print them with the line number.
    2. Print all existing errors during the execution of each pass.
    3. Print each pass separately.
    4. Print at the end a report containing all erroneous lines of source code and the error.

oCode class

- It takes the label, operation, operand, and address vectors as input parameters.
- It computes the object code of each line separately depending on its operation and operand.
- It recognizes the difference between format 3 and format 4.
- It decides whether the code will be pc-relative or base relative.
- It recognizes the difference between direct addressing and indirect addressing.
- It stores the output object codes in a vector.

oProgram class

- It takes the object code, operation, and address vectors as input parameters.
- It calculates the header record and returns it as a string.
- It calculates all text records and returns a vector containing each record.
- It calculates the end record and returns it as a string.

Assumptions

Assumptions

-If there is no start address given with START instructions, we set the default address by 0000.
-Start address less than 7 characters.
-The last line must be END.
- START must be the first line or it can be preceded by a Comment line.

-Label name and variable name must start with a letter.

-All characters are in uppercase.

-In the assembler, if there is an error in pass1, pass2 will not be executed and the symbol table will not be generated.

- if there is an error in pass2, the object program will not be generated.

- EQU any constant gives an absolute.

Sample runs

Sample 1

LINE NO.	ADDRESS	LABEL	OP-CODE	OPERANDS
1		.234567890123456789		
2	000010	AS	START	10
3	000010		BASE	#10
4	000010		LDA	#100
5	000013		COMPR	X,T
6	000015	LOOP	LDCH	BL
7	000018	QQ	LDCH	BLANK
8	00001B	ZFT	EQU	*
9	00001B	MM	EQU	QQ+LOOP
10	00001B	ZZ	EQU	#50
11	00001B		TIX	#100
12	00001E		LDA	#50
13	000021	.		
14	000021	STRI	RESW	300
15	0003A5	BLANK	BYTE	C'ASS'
16	0003A8		END	

> > > > > > > > > > > > > > > > > > > E N D O F P A S S 1

S Y M B O L T A B L E

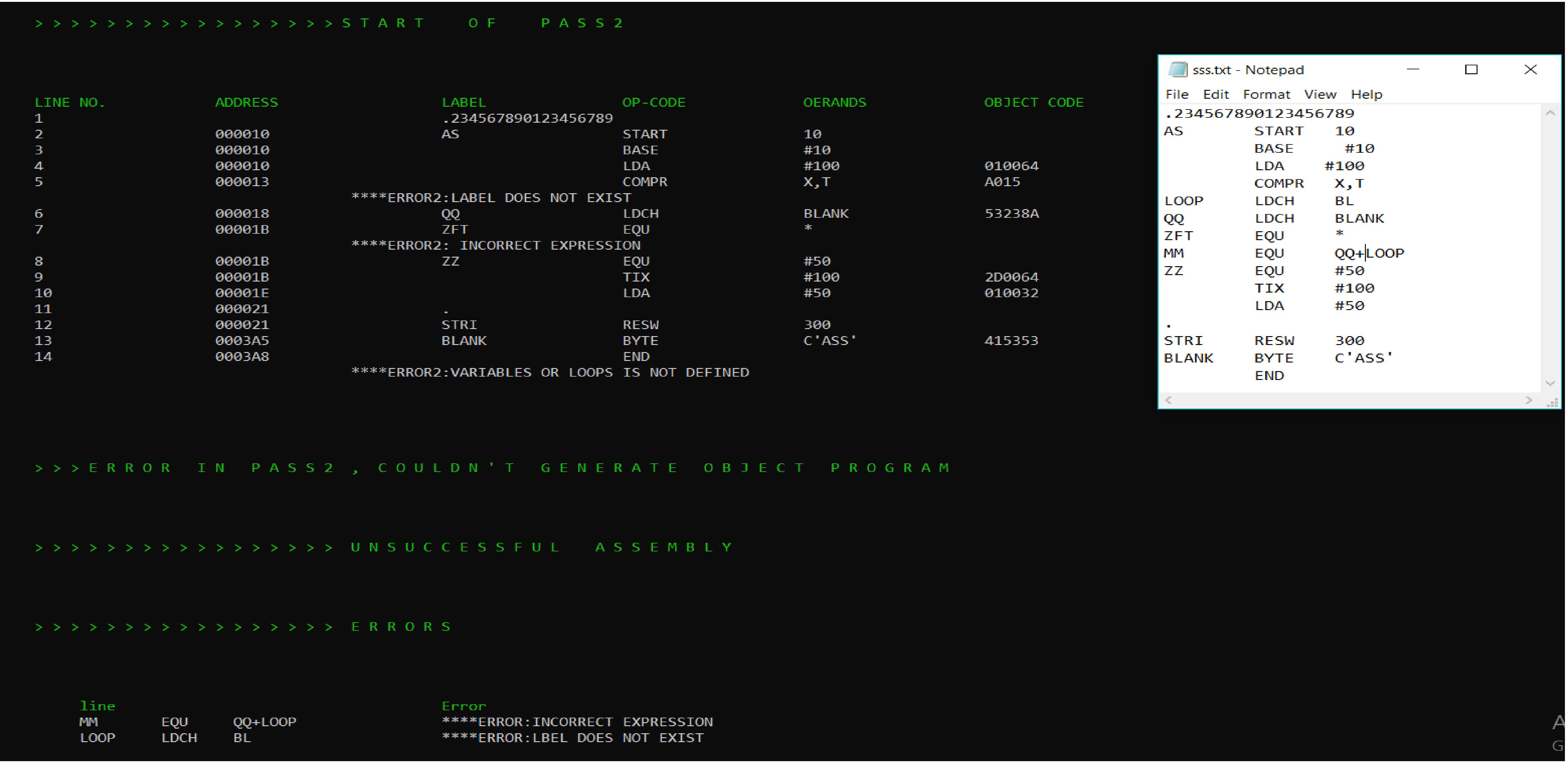
NAME	ADDRESS	ABSOL.RELOC.
STRI	33	relocatable
ZZ	50	absolute
BLANK	933	relocatable
ZFT	27	relocatable
QQ	24	relocatable
LOOP	21	relocatable
AS	16	relocatable

sss.txt - Notepad

File Edit Format View Help

```
.234567890123456789
AS        START   10
          BASE     #10
          LDA    #100
          COMPR   X,T
LOOP      LDCH    BL
QQ        LDCH    BLANK
ZFT       EQU     *
MM        EQU     QQ+LOOP
ZZ        EQU     #50
          TIX     #100
          LDA     #50
.
STRI      RESW    300
BLANK     BYTE    C'ASS'
          END
```

Sample 1



```
> > > > > > > > > > > > > > > > > S T A R T    O F    P A S S 2


LINE NO.            ADDRESS                  LABEL               OP-CODE             OERANDS             OBJECT CODE
1                                            .234567890123456789
2                   000010                   AS                  START               10
3                   000010                                       BASE                #10
4                   000010                                       LDA                 #100                010064
5                   000013                                       COMPR               X,T                 A015
                                   ****ERROR2:LABEL DOES NOT EXIST
6                   000018                   QQ                  LDCH                BLANK               53238A
7                   00001B                   ZFT                 EQU                 *
                                   ****ERROR2: INCORRECT EXPRESSION
8                   00001B                   ZZ                  EQU                 #50
9                   00001B                                       TIX                 #100                2D0064
10                  00001E                                       LDA                 #50                 010032
11                  000021                   .
12                  000021                   STRI                RESW                300
13                  0003A5                   BLANK               BYTE                C'ASS'              415353
14                  0003A8                                       END
                                   ****ERROR2:VARIABLES OR LOOPS IS NOT DEFINED



> > > E R R O R   I N   P A S S 2 ,   C O U L D N ' T   G E N E R A T E   O B J E C T   P R O G R A M



> > > > > > > > > > > > > > > > >  U N S U C C E S S F U L   A S S E M B L Y



> > > > > > > > > > > > > > > > >  E R R O R S



     line                                    Error
     MM       EQU     QQ+LOOP                ****ERROR:INCORRECT EXPRESSION
     LOOP     LDCH    BL                     ****ERROR:LBEL DOES NOT EXIST
```

sss.txt - Notepad

```
.234567890123456789
AS       START   10
         BASE     #10
         LDA    #100
         COMPR   X,T
LOOP     LDCH    BL
QQ       LDCH    BLANK
ZFT      EQU     *
MM       EQU     QQ+LOOP
ZZ       EQU     #50
         TIX     #100
         LDA     #50
.
STRI     RESW    300
BLANK    BYTE    C'ASS'
         END
```

Sample 2

LINE NO.	ADDRESS	LABEL	OP-CODE	OPERANDS
1	000000	COPY	START	0
2	000000	FIRST	STL	RETADR
3	000003		LDB	#LENGTH
4	000006		BASE	LENGTH
5	000006	CLOOP	+JSUB	RDREC
6	00000A		LDA	LENGTH
7	00000D		COMP	#0
8	000010		JEQ	ENDFIL
9	000013		+JSUB	WRREC
10	000017		J	CLOOP
11	00001A	ENDFIL	LDA	EOF
12	00001D		STA	BUFFER
13	000020		LDA	#3
14	000023		STA	LENGTH
15	000026		+JSUB	WRREC
16	00002A		J	@RETADR
17	00002D	EOF	BYTE	C'EOF'
18	000030	RETADR	RESW	1
19	000033	BUFFER	RESB	4096
20	001033	.		
21	001033	.		
22	001033	.		
23	001033	RDREC	CLEAR	X
24	001035		CLEAR	A
25	001037		CLEAR	S
26	001039		+LDT	#4096
27	00103D	RLOOP	TD	INPUT
28	001040		JEQ	RLOOP
29	001043		RD	INPUT
30	001046		COMPR	A,S
31	001048		JEQ	EXIT
32	00104B		STCH	BUFFER,X
33	00104E		TIXR	T
34	001050		JLT	RLOOP
35	001053	EXIT	STX	LENGTH
36	001056		RSUB	
37	001059	INPUT	BYTE	X'F1'
38	00105A	.		
39	00105A	.		
40	00105A	.		
41	00105A	WRREC	CLEAR	X
42	00105C		LDT	LENGTH
43	00105F	WLOOP	TD	OUTPUT
44	001062		JEQ	WLOOP
45	001065		LDCH	BUFFER,X
46	001068		WD	OUTPUT
47	00106B		TIXR	T
48	00106D		JLT	WLOOP

```
COPY START 0
FIRST STL RETADR
   LDB #LENGTH
   BASE LENGTH
CLOOP +JSUB RDREC
   LDA  LENGTH
   COMP  #0
   JEQ  ENDFIL
   +JSUB  WRREC
  J CLOOP
ENDFIL LDA EOF
   STA  BUFFER
   LDA #3
   STA  LENGTH
  +JSUB WRREC
   J @RETADR
EOF  BYTE  C'EOF'
RETADR RESW 1
BUFFER RESB 4096
.
RDREC  CLEAR X
   CLEAR A
   CLEAR S
   +LDT  #4096
RLOOP TD INPUT
   JEQ  RLOOP
  RD INPUT
   COMPR A,S
   JEQ EXIT
  STCH BUFFER,X
   TIXR T
   JLT RLOOP
EXIT  STX  LENGTH
   RSUB
INPUT BYTE X'F1'
.
WRREC CLEAR X
   LDT LENGTH
WLOOP  TD OUTPUT
   JEQ WLOOP
   LDCH BUFFER,X
   WD OUTPUT
   TIXR  T

 JLT  WLOOP
  RSUB
OUTPUT  BYTE   X'05'
     END  FIRST
```

Sample 2

49	001070		RSUB	
50	001073	OUTPUT	BYTE	X'05'
51	001074		END	FIRST

> > > > > > > > > > > > > > > > > > E N D O F P A S S 1

S Y M B O L T A B L E

NAME	ADDRESS	ABSOL.RELOC.
OUTPUT	4211	relocatable
WLOOP	4191	relocatable
INPUT	4185	relocatable
COPY	0	relocatable
FIRST	0	relocatable
CLOOP	6	relocatable
ENDFIL	26	relocatable
RDREC	4147	relocatable
EOF	45	relocatable
RETADR	48	relocatable
BUFFER	51	relocatable
WRREC	4186	relocatable
RLOOP	4157	relocatable
EXIT	4179	relocatable

> > > > > > > > > > > > > > > > > > S T A R T O F P A S S 2

LINE NO.	ADDRESS	LABEL	OP-CODE	OERANDS	OBJECT CODE
1	000000	COPY	START	0	
2	000000	FIRST	STL	RETADR	17202D
3	000003		LDB	#LENGTH	692000
		****ERROR2:LABEL DOES NOT EXIST			
4	000006	CLOOP	+JSUB	RDREC	4B101033
		****ERROR2:LABEL DOES NOT EXIST			
5	00000D		COMP	#0	290000
6	000010		JEQ	ENDFIL	332007
7	000013		+JSUB	WRREC	4B10105A
8	000017		J	CLOOP	3F2FEC
9	00001A	ENDFIL	LDA	EOF	032010
10	00001D		STA	BUFFER	0F2013
11	000020		LDA	#3	010003
		****ERROR2:LABEL DOES NOT EXIST			
12	000026		+JSUB	WRREC	4B10105A
13	00002A		J	@RETADR	3E2003
14	00002D	EOF	BYTE	C'EOF'	454f46
15	000030	RETADR	RESW	1	
16	000033	BUFFER	RESB	4096	
17	001033	.			
18	001033	.			
19	001033	.			
20	001033	RDREC	CLEAR	X	B410
21	001035		CLEAR	A	B400
22	001037		CLEAR	S	B440
23	001039		+LDT	#4096	75101000
24	00103D	RLOOP	TD	INPUT	E32019
25	001040		JEQ	RLOOP	332FFA
26	001043		RD	INPUT	DB2013
27	001046		COMPR	A,S	A004
28	001048		JEQ	EXIT	332008
29	00104B		STCH	BUFFER,X	57CFBF
30	00104E		TIXR	T	B850
31	001050		JLT	RLOOP	3B2FEA
		****ERROR2:LABEL DOES NOT EXIST			
32	001056		RSUB		4F0000
33	001059	INPUT	BYTE	X'F1'	F1
34	00105A	.			
35	00105A	.			
36	00105A	.			
37	00105A	WRREC	CLEAR	X	B410
		****ERROR2:LABEL DOES NOT EXIST			
38	00105F	WLOOP	TD	OUTPUT	E32011

Sample 2

```
39          001062                          JEQ         WLOOP       332FFA
40          001065                          LDCH        BUFFER,X    53CFBF
41          001068                          WD          OUTPUT      DF2008
42          00106B                          TIXR        T           B850
43          00106D                          JLT         WLOOP       3B2FEF
44          001070                          RSUB                    4F0000
45          001073          OUTPUT          BYTE        X'05'       05
46          001074                          END         FIRST
                    ****ERROR2:VARIABLES OR LOOPS IS NOT DEFINED
                    ****ERROR2:VARIABLES OR LOOPS IS NOT DEFINED
                    ****ERROR2:VARIABLES OR LOOPS IS NOT DEFINED
                    ****ERROR2:VARIABLES OR LOOPS IS NOT DEFINED
                    ****ERROR2:VARIABLES OR LOOPS IS NOT DEFINED

>>>ERROR IN PASS2 , COULDN'T GENERATE OBJECT PROGRAM

>>>>>>>>>>>>>>>>>> UNSUCCESSFUL  ASSEMBLY

>>>>>>>>>>>>>>>>>> ERRORS

    line                            Error
      BASE LENGTH                   ****ERROR:LBEL DOES NOT EXIST
      LDA  LENGTH                   ****ERROR:LBEL DOES NOT EXIST
      STA  LENGTH                   ****ERROR:LBEL DOES NOT EXIST
    EXIT  STX  LENGTH               ****ERROR:LBEL DOES NOT EXIST
      LDT LENGTH                    ****ERROR:LBEL DOES NOT EXIST
```

Sample 3

LINE NO.	ADDRESS	LABEL	OP-CODE	OPERANDS
1		.234567890123456789		
2	000010	AS	START	10
3	000010		BASE	#10
4	000010		LDA	#100
5	000013		COMPR	X,T
6	000015	LOOP	LDCH	BLANK
7	000018	QQ	LDCH	BLANK
8	00001B	ZFT	EQU	*
9	00001B	MM	EQU	QQ-LOOP
10	00001B	ZZ	EQU	#50
11	00001B		TIX	#100
12	00001E		LDA	#50
13	000021	.		
14	000021	STRI	RESW	300
15	0003A5	BLANK	BYTE	C'ASS'
16	0003A8		END	

>>>>>>>>>>>>>>>>>>>END OF PASS1

SYMBOL TABLE

NAME	ADDRESS	ABSOL.RELOC.
STRI	33	relocatable
ZZ	50	absolute
BLANK	933	relocatable
ZFT	27	relocatable
MM	3	absolute
QQ	24	relocatable
LOOP	21	relocatable
AS	16	relocatable

sss.txt - Notepad

File Edit Format View Help

```
.234567890123456789
AS        START   10
          BASE     #10
          LDA    #100
          COMPR   X,T
LOOP      LDCH    BLANK
QQ        LDCH    BLANK
ZFT       EQU     *
MM        EQU     QQ-LOOP
ZZ        EQU     #50
          TIX     #100
          LDA     #50
.
STRI      RESW    300
BLANK     BYTE    C'ASS'
          END
```

Sample 3

```
>>>>>>>>>>>>>>>>>>>START   OF   PASS2


LINE NO.        ADDRESS         LABEL           OP-CODE         OERANDS         OBJECT CODE
1                               .234567890123456789
2               000010          AS              START           10
3               000010                          BASE            #10
4               000010                          LDA             #100            010064
5               000013                          COMPR           X,T             A015
6               000015          LOOP            LDCH            BLANK           53238D
7               000018          QQ              LDCH            BLANK           53238A
8               00001B          ZFT             EQU             *
9               00001B          MM              EQU             QQ-LOOP
10              00001B          ZZ              EQU             #50
11              00001B                          TIX             #100            2D0064
12              00001E                          LDA             #50             010032
13              000021          .
14              000021          STRI            RESW            300
15              0003A5          BLANK           BYTE            C'ASS'          415353
16              0003A8                          END


        OBJECT   PROGRAM


    HAS         0003A8
    T0000100B010064A01553238D53238A
    T00001B062D0064010032
    T0003A503415353E000010


>>>>>>>>>>>>>>>>>>>END   OF   PASS2


>>>>>>>>>>>>>>>>>>>> SUCCESSFUL   ASSEMBLY
```

sss.txt - Notepad

File Edit Format View Help

```
.234567890123456789
AS      START   10
        BASE     #10
        LDA    #100
        COMPR   X,T
LOOP    LDCH    BLANK
QQ      LDCH    BLANK
ZFT     EQU     *
MM      EQU     QQ-LOOP
ZZ      EQU     #50
        TIX     #100
        LDA     #50
.
STRI    RESW    300
BLANK   BYTE    C'ASS'
        END
```