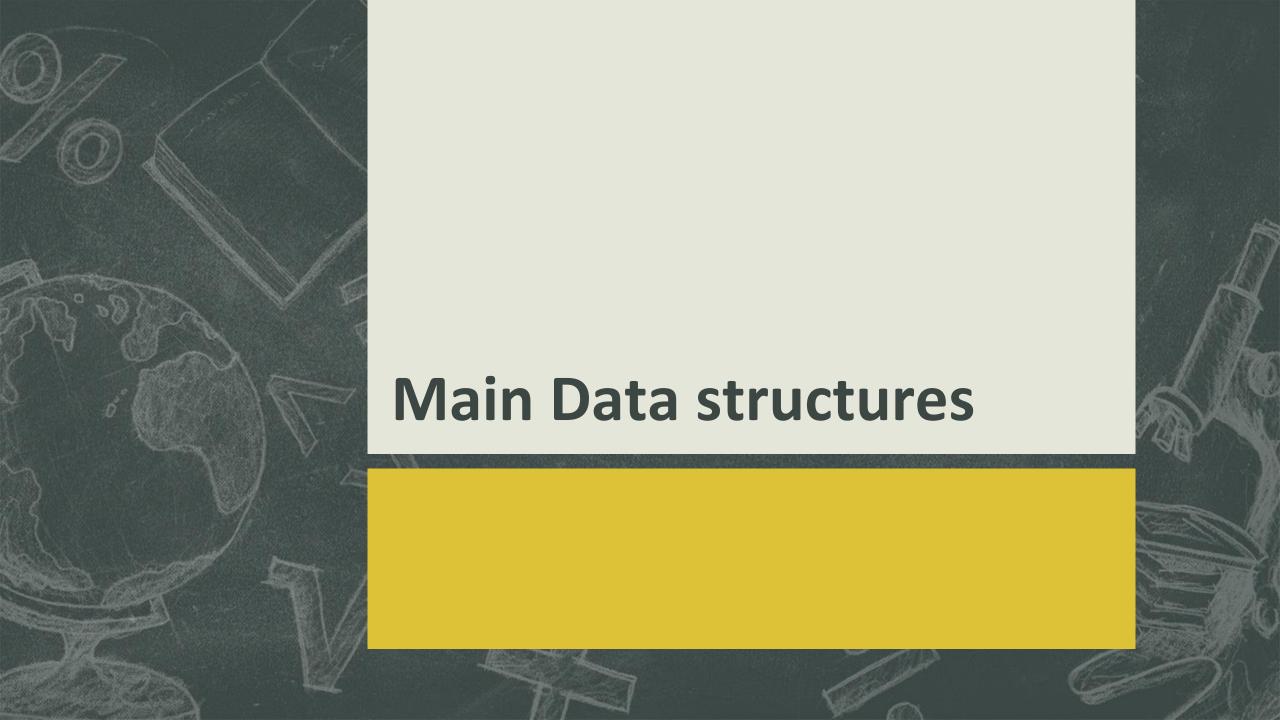




- 1. To execute pass1 you should enter pass1<source-file-name>.
- 2. To execute pass2 you should enter pass2<source-file-name>.
- 3. To execute the whole assembly, you should enter assemble<source-file-name>.
- 4. The source file for the main program is named assemble.cpp.
- 5. Handling phase 2 errors 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.



- 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.

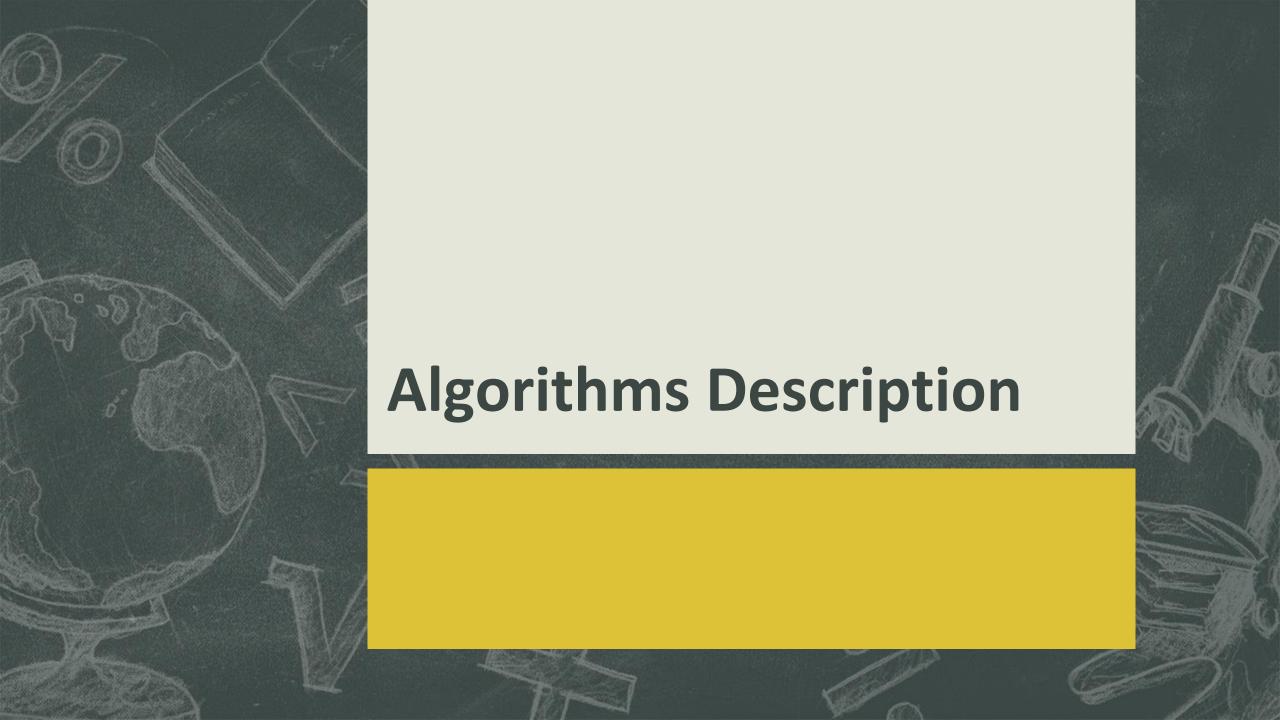


Maps and vectors

- 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



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

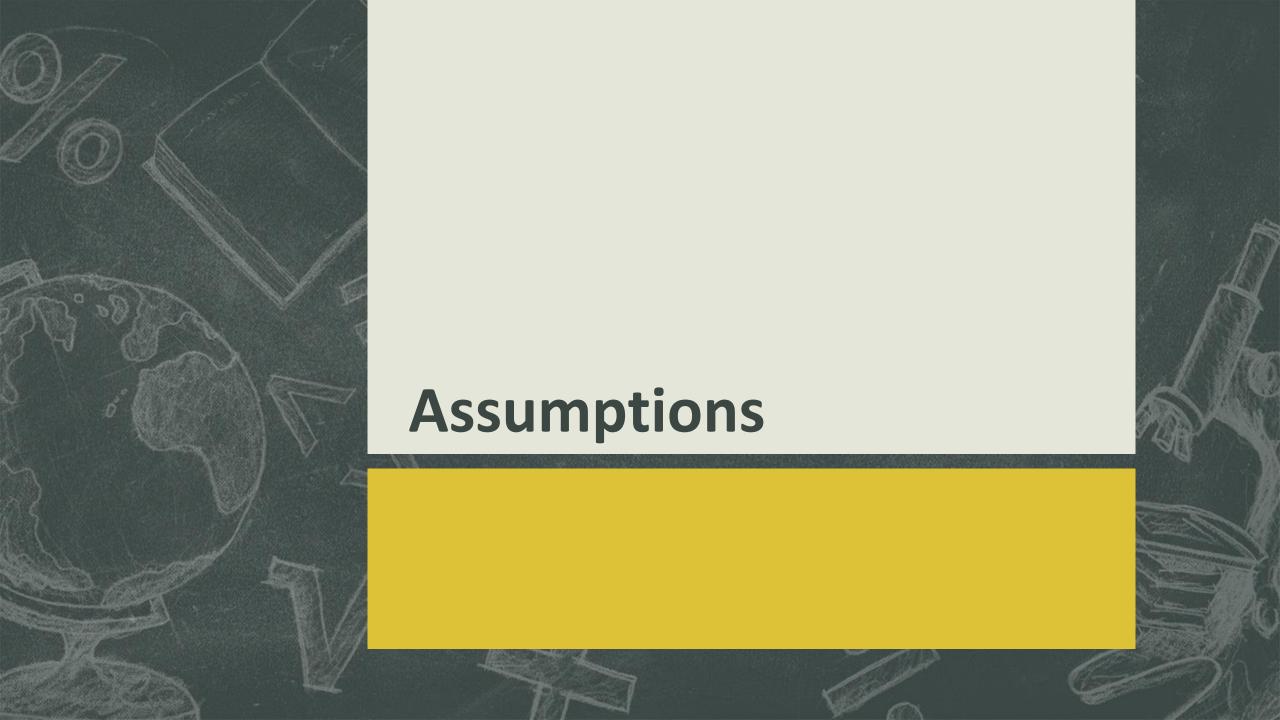
- In case of wanting to print pass1 only or pass2 only:
 - 1. They take the vectors that contain the Mnemonic Op-codes, operands, addresses, labels, object codes, the map that contains the symbol table and print them with the line number.
 - 2. Print all existing errors.
- In case of wanting to print 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.

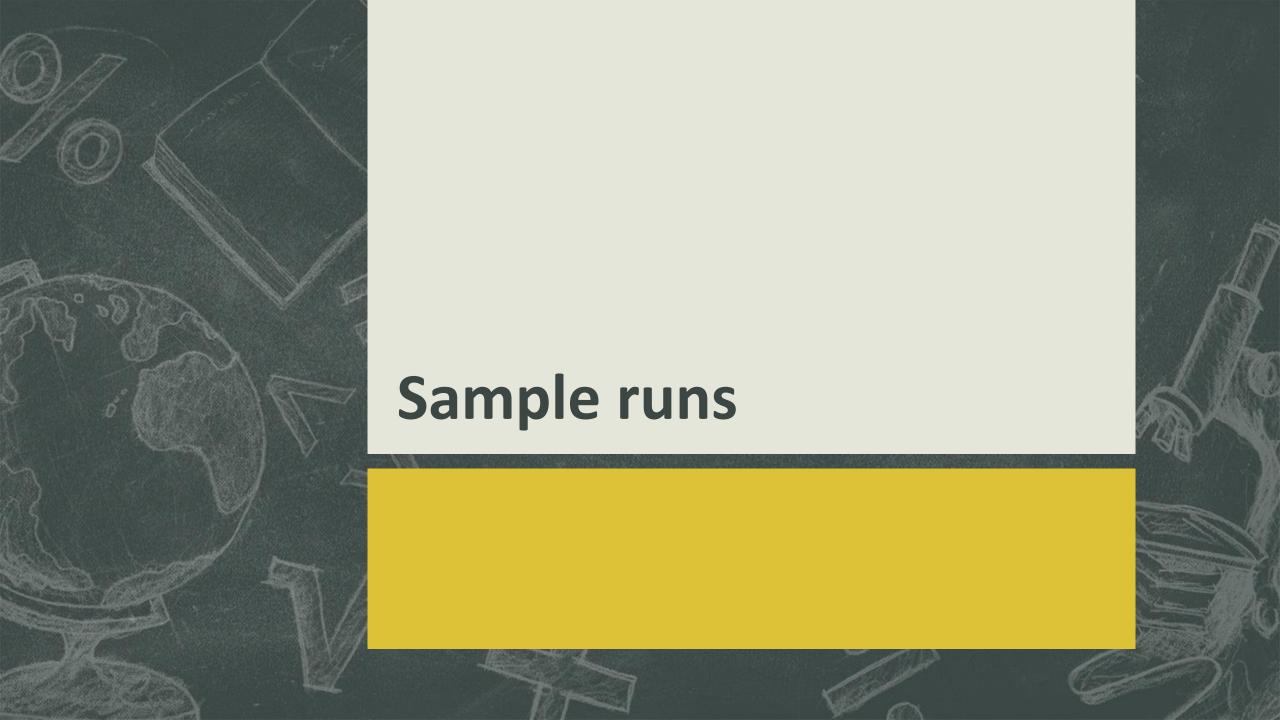
oProgram class

- 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

- -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.

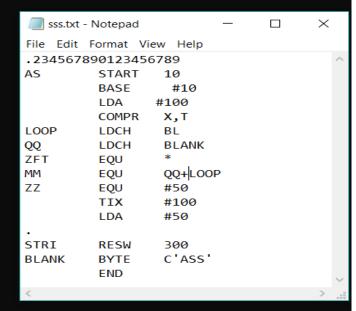


LINE NO.	ADDRESS	LABEL	OP-CODE	OPERANDS	sss.txt	- Notepad	_	×
1		.234567890123	456789		File Edit	Format V	iew Help	
2	000010	AS	START	10	.234567	89012345	6789	^
3	000010		BASE	#10	AS	START	10	
4	000010		LDA	#100		BASE	#10	
5	000013		COMPR	X,T		LDA	#100	
6	000015	LOOP	LDCH	BL		COMPR	X, T	
7	000018	QQ_	LDCH	BLANK	LOOP	LDCH	BL	
8	00001B	ZFT	EQU	*				
9	00001B	MM	EQU	QQ+LOOP	QQ	LDCH	BLANK *	
10	00001B	ZZ	EQU	#50	ZFT	EQU		
11	00001B		TIX	#100	MM	EQU	QQ+LOOP	
12	00001E		LDA	#50	ZZ	EQU	#50	
13	000021	· CTDT	DECH	300		TIX	#100	
14	000021	STRI	RESW	300		LDA	#50	
15 16	0003A5 0003A8	BLANK	BYTE END	C'ASS'				
10	0003A8		END		STRI	RESW	300	
					BLANK	BYTE	c'ASS'	
	>>>>> E N	ID OF PAS	S 1			END		
			3 1					
					<			> .::

SYMBOL TABLE

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

LINE NO.	ADDRESS	LABEL	OP-CODE	OERANDS	OBJECT CODE
1		.234567890123	456789		
2	000010	AS	START	10	
3	000010		BASE	#10	
4	000010		LDA	#100	010064
5	000013		COMPR	X,T	A015
		****ERROR2:LABEL DOES N	OT EXIST		
6	000018	QQ	LDCH	BLANK	53238A
7	00001B	ZFT	EQU	*	
		****ERROR2: INCORRECT E	XPRESSION		
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 DEFINE	D	



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

>>>>>>>>>>>

>>>>> ERRORS

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

LINE NO. ADDRESS LABEL OP-CODE OPERANDS					
2	LINE NO.	ADDRESS	LABEL	OP-CODE	OPERANDS
2	1	999999	CODY	START	a
1					
BASE			11131		
S					
6 00000A 7 00000D	5		CLOOP		
7 00000D COMP #0 8 000013 JEQ ENDFIL 9 000013 JSUB WRRCC 10 000017 J CLOOP 11 00001A ENDFIL LDA EOF 12 00001D STA BUFFER 13 000020 LDA #3 14 000023 STA LENTH 15 000026 J WRREC 16 00002A J QRETADR 17 000030 RETADR RESW L 19 000033 BUFFER RESW 4 20 001033 . . 21 001033 . . 22 001033 . . 23 001033 . . 24 001033 . . 25 001037 CLEAR X 26 001037 CLEAR X			2230.		
S					
9					
10					
11					
12			ENDETL		
13					
14 000023 STA LENGTH 15 000026 +JSUB WRREC 16 00002A J @RETADR 17 000020 EOF BYTE C'EOF' 18 000030 RETADR RESW 1 19 000033 BUFFER RESB 4096 20 001033 - - 21 001033 - - 22 001033 - - 23 001033 - CLEAR X 24 001035 CLEAR X 25 001037 CLEAR S 26 001037 RLOOP TD INPUT 28 001040 BLOOP TD INPUT 29 001043 COMPR A,S 31 001048 COMPR A,S 31 001048 TIXR T 32 001048 TIXR T 33 001048 STCH BUFER,X 35 001050 <td< td=""><td></td><td></td><td></td><td></td><td></td></td<>					
15					
16 00002A EOF BYTE C'EOF' 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 A 26 001039 RLOOP TD INPUT 28 001040 JEQ RLOOP 29 001043 RD INPUT 30 001046 COMPR A,S 31 001048 JEQ EXIT 32 00104B TIXR T 34 001050 EXIT RSUB 37 001053 EXIT STX LENGTH 38 00105A . . . 39	15				
17					
19			EOF	BYTE	
19	18	000030	RETADR	RESW	1
21					4096
21	20	001033			
23		001033			
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 X'F1' 37 001059 INPUT BYTE X'F1' 38 00105A . . 40 00105A . . 41 00105A . LDT LENGTH 43 00105F WRREC CLEAR X 41 00105A . LDT LENGTH 43 00105F	22	001033			
25	23	001033	RDREC	CLEAR	
26	24	001035		CLEAR	Α
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 VFI X'F1' 38 00105A . X'F1' X'F1' 39 00105A . LDT LENGTH 40 00105A . LDT LENGTH 42 00105C LDT LENGTH 43 00105F WLOOP TD OUTPUT 44 001062 JEQ WLOOP 45 001065 WLOOP TUCH BUFFER,X 46 001068 WD OUTPUT 47 001068 TIXR TIXR	25	001037		CLEAR	S
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 STYFI X'F1' 38 00105A . X'F1' X'F1' 40 00105A . LDT LENGTH 41 00105A . LDT LENGTH 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	26	001039		+LDT	#4096
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 X'F1' 37 001059 INPUT BYTE X'F1' 38 00105A . X'F1' 40 00105A . LDT LENGTH 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 001068 TIXR T	27	00103D	RLOOP		INPUT
30					RLOOP
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 . 40 00105A . 41 00105A . 42 00105C LDT LENGTH 43 00105F WLOOP TD OUTPUT 44 001062 JEQ WLOOP 45 001065 LDCH BUFFER, X 46 001068 WD OUTPUT 47 001068 TIXR T					
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 . 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 001068 TIXR T					
33 00104E TIXR T 34 001050 JLT RLOOP 35 001053 EXIT STX LENGTH 36 001056 RSUB T 37 001059 INPUT BYTE X'F1' 38 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 001068 TIXR T					
34 001050 JLT RLOOP 35 001053 EXIT STX LENGTH 36 001056 RSUB TINPUT BYTE X'F1' 38 00105A . TINPUT STX LENGTH 39 00105A . TINPUT STX LENGTH 40 00105A . LDT LENGTH 41 00105C LDT LENGTH 42 00105C LDT LENGTH 43 00105F WLOOP TD OUTPUT 44 001062 JEQ WLOOP 45 001065 LDCH BUFFER, X 46 001068 WD OUTPUT 47 001068 TIXR T					
35 001053 EXIT STX LENGTH 36 001056 RSUB X'F1' 37 001059 INPUT BYTE X'F1' 38 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					
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					
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			EXIT		LENGTH
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					wie.i
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				BAIF	X.F1.
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					
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					
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			·	CLEAR	V
43 00105F WLOOP TD OUTPUT 44 001062 JEQ WLOOP 45 001065 LDCH BUFFER,X 46 001068 WD OUTPUT 47 00106B TIXR T			WRREC		
44 001062 JEQ WLOOP 45 001065 LDCH BUFFER,X 46 001068 WD OUTPUT 47 00106B TIXR T			LIL COD		
45 001065 LDCH BUFFER,X 46 001068 WD OUTPUT 47 00106B TIXR T			WEOOP		
46 001068 WD OUTPUT 47 00106B TIXR T					
47 00106B TIXR T					
40 JET WEODP					
	40	001000		JET	WEOOI

COPY START Ø 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

49 50		070 073	OUTPUT	RSUB BYTE X	('05'	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	>>>>>>>	> S T A R T 0 F	PASS2		
51		.074	001101		IRST						
31	001			END I	11131						
						LINE NO.	ADDRESS	LABEL	OP-CODE	OERANDS	OBJECT CODE
> >		>>>> E N D	0 F P A S S 1			1	000000	COPY	START	0 0	OBJECT CODE
						2	000000	FIRST	STL	RETADR	17202D
						3	000003		LDB	#LENGTH	692000
	SYMB	OL TABLE						****ERROR2:LABEL DOE			
						4	000006	CLOOP	+JSUB	RDREC	4B101033
	NAME	ADDRESS	ABSOL.RELOC.			_		****ERROR2:LABEL DOE			
						5	00000D		COMP	#0	290000
	OUTPUT	4211	relocatable			6 7	000010 000013		JEQ +JSUB	ENDFIL WRREC	332007 4B10105A
	331131	4222	· clocacable			8	000013		J	CL00P	3F2FEC
	WLOOP	4191	relocatable			9	000017 00001A	ENDFIL	LDA	EOF	032010
	WLOOP	4191	relocatable			10	00001D	21131 12	STA	BUFFER	0F2013
	THOUT	44.05	3			11	000020		LDA	#3	010003
	INPUT	4185	relocatable					****ERROR2:LABEL DOE	S NOT EXIST		
						12	000026		+JSUB	WRREC	4B10105A
	COPY	0	relocatable			13	00002A		J	@RETADR	3E2003
						14	00002D	EOF	BYTE	C'EOF'	454f46
	FIRST	0	relocatable			15	000030	RETADR	RESW	1	
						16	000033	BUFFER	RESB	4096	
	CL00P	6	relocatable			17 18	001033 001033				
						19	001033				
	ENDFIL	26	relocatable			20	001033	RDREC	CLEAR	Χ	B410
	21401 12	20	refocatable			21	001035	RENEC	CLEAR	A	B400
	RDREC	4147	relocatable			22	001037		CLEAR	S	B440
	RDREC	414/	relocatable			23	001039		+LDT	#4096	75101000
						24	00103D	RLOOP	TD	INPUT	E32019
	EOF	45	relocatable			25	001040		JEQ	RLOOP	332FFA
						26	001043		RD	INPUT	DB2013
	RETADR	48	relocatable			27	001046		COMPR	A,S	A004
						28	001048		JEQ	EXIT	332008
	BUFFER	51	relocatable			29	00104B		STCH	BUFFER,X	57CFBF
						30 31	00104E 001050		TIXR JLT	T RLOOP	B850 3B2FEA
	WRREC	4186	relocatable			31	001030	****ERROR2:LABEL DOE		KLOUP	SDZFEA
						32	001056	ENNONZ: LABLE DOL	RSUB		4F0000
	RLOOP	4157	relocatable			33	001059	INPUT	BYTE	X'F1'	F1
	KLOOF	4137	relocatable			34	00105A				
	LALL	4170				35	00105A				
	EXIT	4179	relocatable			36	00105A				
						37	00105A	WRREC	CLEAR	X	B410
								****ERROR2:LABEL DOE			
						38	00105F	WLOOP	TD	OUTPUT	E32011

```
001062
                                                       JEQ
                                                                        WLOOP
                                                                                         332FFA
                001065
                                                                        BUFFER,X
                                                                                         53CFBF
40
                                                       LDCH
41
                                                                        OUTPUT
                 001068
                                                       WD
                                                                                         DF2008
42
                 00106B
                                                       TIXR
                                                                                         B850
                 00106D
                                                                        WLOOP
                                                                                         3B2FEF
                                                       RSUB
44
                                                                                         4F0000
                 001070
                 001073
                                      OUTPUT
                                                       BYTE
                                                                        X'05'
                                                                                         05
46
                 001074
                                                                        FIRST
                                                       END
                              ****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
>>>>>>>>>>>>
>>>>>>> ERRORS
                                      ****ERROR:LBEL DOES NOT EXIST
     BASE LENGTH
     LDA LENGTH
                                      ****ERROR:LBEL DOES NOT EXIST
                                      ****ERROR:LBEL DOES NOT EXIST
     STA LENGTH
    EXIT STX LENGTH
                                      ****ERROR:LBEL DOES NOT EXIST
     LDT LENGTH
                                      ****ERROR:LBEL DOES NOT EXIST
```

LINE	NO.	ADDRESS	LABEL	OP-CODE	OPERANDS				
1 2 3 4 5		000010 000010 000010 000013	. 234567890123456789 AS	START BASE LDA COMPR	10 #10 #100 X,T	sss.txt - File Edit .2345678	Format Vie		×
6 7 8 9 10 11 12		000015 000018 00001B 00001B 00001B 00001B 00001E 00001E	LOOP QQ ZFT MM ZZ	LDCH LDCH EQU EQU TIX LDA	BLANK BLANK * QQ-LOOP #50 #100 #50	AS LOOP QQ ZFT	START BASE LDA COMPR LDCH LDCH EQU	10 #10 #100 X,T BLANK BLANK	ı
14 15 16		000021 0003A5 0003A8	STRI BLANK	RESW BYTE END	300 C'ASS'	MM ZZ	EQU EQU TIX LDA	QQ-LOOP #50 #100 #50	
>>		>>>>>	OF PASS1			STRI BLANK	RESW BYTE END	300 C'ASS'	~
	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						

LINE NO. ADDRESS 1 2	LABEL .23456789012 AS LOOP QQ ZFT MM ZZ	START BASE LDA COMPR LDCH LDCH EQU EQU TIX	OERANDS 10 #10 #100 X,T BLANK BLANK * QQ-LOOP	0BJECT CODE 010064 A015 53238D 53238A	
000010 000010 000013 000015 000018 00001B 00001B 00001B 1 00001B 2 00001B 2 00001E 3 000021 4 000021	LOOP QQ ZFT MM ZZ	BASE LDA COMPR LDCH LDCH EQU EQU EQU TIX	#10 #100 X,T BLANK BLANK * QQ-LOOP	A015 53238D	
000010 000013 000015 000018 00001B 00001B 00001B 1 00001B 2 00001E 3 000021 4 000021	QQ ZFT MM ZZ	LDA COMPR LDCH LDCH EQU EQU EQU TIX	#100 X,T BLANK BLANK * QQ-LOOP	A015 53238D	
000013 000015 000018 00001B 00001B 00001B 00001B 2 00001E 3 000021 4 000021	QQ ZFT MM ZZ	COMPR LDCH LDCH EQU EQU EQU TIX	X,T BLANK BLANK * QQ-LOOP	A015 53238D	
000015 000018 00001B 00001B 00001B 00001B 000001E 0000021 0000021	QQ ZFT MM ZZ	LDCH EQU EQU TIX	BLANK BLANK * QQ-LOOP	53238D	
000018 00001B 00001B 00001B 00001B 00001E 000021 000021	QQ ZFT MM ZZ	EQU EQU TIX	BLANK * QQ-LOOP		
00001B 00001B 00001B 00001B 00001E 000021 000021	ZFT MM ZZ	LIX EĞN EĞN	* QQ-LOOP	53238A	
00001B 00001B 00001B 00001E 000021 000021	MM ZZ	EQU TIX	QQ-LOOP		
00001B 00001B 00001E 000021 000021	ZZ	EQU TIX			
00001B 00001E 000021 000021 0003A5		TIX			
00001E 000021 000021 0003A5			#50		
000021 000021 0003A5		1.00	#100	2D0064	
000021 0003A5		LDA	#50	010032	
0003A5		DECH	300		
	STRI	RESW	300	445252	
0003A8	BLANK	BYTE END	C'ASS'	415353	
овјест рк	OGRAM		A	BASE #10	,
HAS 0003A8				LDA #100 COMPR X,T OOP LDCH BLANK	
T0000100B010064A01553238	D53238A		Q		
T00001B062D0064010032				FT EQU *	
T0003A503415353E000010			M		
			Z	•	
				TIX #100	
				LDA #50	
				TRI RESW 300	
	>>END OF PAS	5.2	В	LANK BYTE C'ASS'	
				END	