Programming Assignment #1

Assignment Description

Write an SIC assembler that reads an SIC assembly program, translates SIC statements into their machine code equivalents, and generates an object file.

Goals

1. Get familiar with C programming language.
2. Learn to use the I/O facilities and library functions provided by standard C.
3. Get experience with the system-level programming.
4. Get familiar with separate compilation, make utility, and C debugger.

Guidelines:

1. The score for a programming assignment includes demonstration in person (70%) and a report (30%). You have to demonstrate your program in person. For a two-member team, two programming assignments need to be demonstrated by different members. You will get 15% bonus if you succeed in demonstrating your program (without any obvious bugs) on November 19 noon, or earlier. On the other hand, you will be 15% penalty if you failed in demonstrating your program or handing in your report on December 24 noon, or earlier.
    1) Your score = X * 70% + Y * 30%
    2) (11/4) In-class demonstration = X * 70% * 115% + Y * 30%
    3) Late demonstration = X * 70% * 85% + Y * 30%
    4) Late demonstration + late report = X * 70% * 85% + Y * 30% * 85%
2. You need to upload your report to www.elearn.ndhu.edu.tw
    1) Your report is suggested to include the following elements:
        i. A cover page
        ii. Brief description of the assignment problem
        iii. Highlights on how you write the program
        iv. Program listing
        v. Test run snapshots
        vi. Discussion
    2) Make your report in a pdf file, and upload to www.elearn.ndhu.edu.tw.
    3) The upload link will be suspended at the due time for a while for recording on-time hand-ins, and will be reopened for accepting late hand-ins. There will be a 10% penalty for lateness. (S = X * 70% + Y * 30% * 85%)

Programming Assignment #1 Addendum

Assembler Directives

START, END, BYTE, WORD, RESB, and RESW.

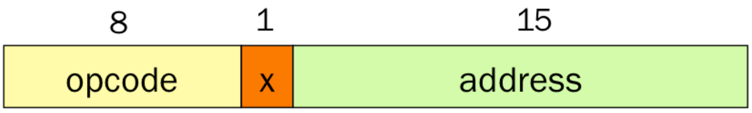
Format of SIC Assembly Program

1) Each line has 4 fields:
   i. (Optional) Label (up to 6 characters)
   ii. Operation
   iii. (Optional) Operands (no blank allowed)
   iv. (Optional) Comment
2) If the first character is a ".", the whole line is a comment.
3) There is no blank line in an SIC assembly program source file, except for the last line.

SIC OPCODE Table

Mnemonic	Opcode	Mnemonic	Opcode	Mnemonic	Opcode
ADD	18	AND	40	COMP	28
DIV	24	J	3C	JEQ	30
JGT	34	JLT	38	JSUB	48
LDA	00	LDCH	50	LDL	08
LDX	04	MUL	20	OR	44
RD	D8	RSUB	4C	STA	0C
STCH	54	STL	14	STSW	E8
STX	10	SUB	1C	TD	E0
TIX	2C	WD	DC		

SIC Instruction Format



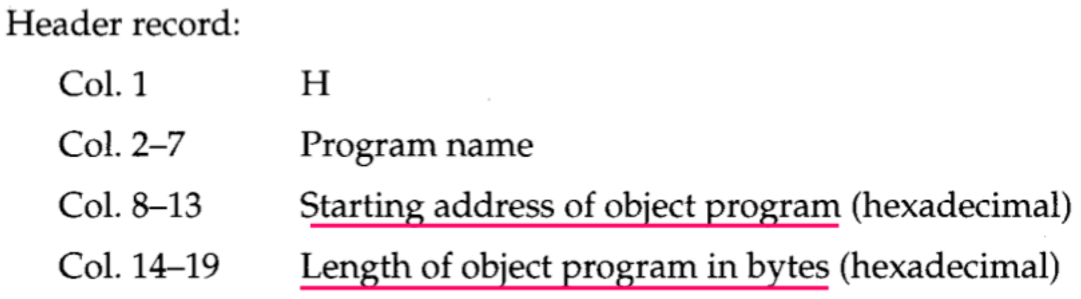
Mode	Indication	Target Address
Direct	x = 0	TA = address
Indexed	x = 1	TA = address + (X)

Example Assembly Program

```
COPY   START  1000
FIRST  STL    RETADR
CLOOP  JSUB   RDREC
       LDA    LENGTH
       COMP   ZERO
       JEQ    ENDFIL
       JSUB   WRREC
       J      CLOOP
ENDFIL LDA    EOF
       STA    BUFFER
       LDA    THREE
       STA    LENGTH
       JSUB   WRREC
       LDL    RETADR
       RSUB
EOF    BYTE   C’EOF’
THREE  WORD   3
ZERO   WORD   0
RETADR RESW   1
LENGTH RESW   1
BUFFER RESB   4096
.
.      SUBROUTINE TO READ RECORD INTO BUFFER
.
RDREC  LDX    ZERO
       LDA    ZERO
RLOOP  TD     INPUT
       JEQ    RLOOP
       RD     INPUT
       COMP   ZERO
       JEQ    EXIT
       STCH   BUFFER,X
       TIX    MAXLEN
       JLT    RLOOP
EXIT   STX    LENGTH
       RSUB
INPUT  BYTE   X’F1’
```

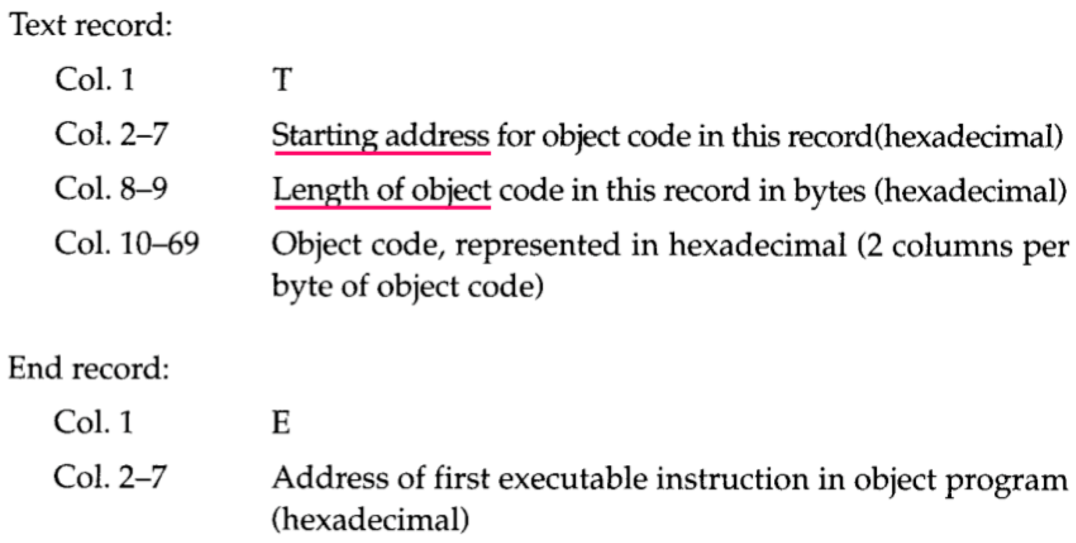
```
MAXLEN WORD   4096
.
.      SUBROUTINE TO WRITE RECORD FROM BUFFER
.
WRREC  LDX    ZERO
WLOOP  TD     OUTPUT
       JEQ    WLOOP
       LDCH   BUFFER,X
       WD     OUTPUT
       TIX    LENGTH
       JLT    WLOOP
       RSUB
OUTPUT BYTE   X'05'
       END    FIRST
```

Types of Record in SIC Object Code



Header record:

Col. 1	H
Col. 2–7	Program name
Col. 8–13	Starting address of object program (hexadecimal)
Col. 14–19	Length of object program in bytes (hexadecimal)



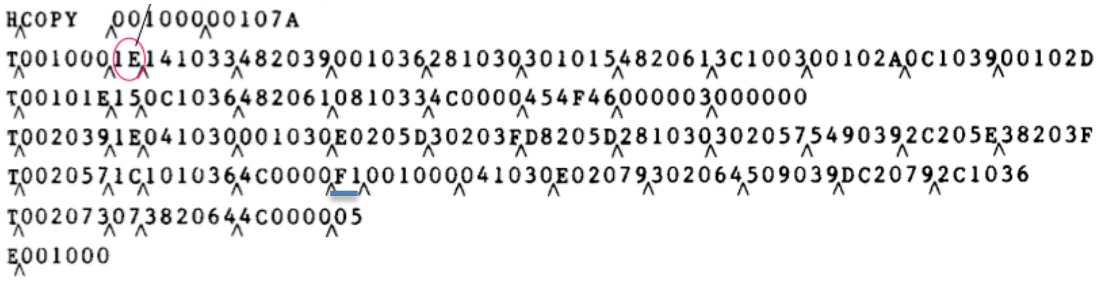
Text record:

Col. 1	T
Col. 2–7	Starting address for object code in this record(hexadecimal)
Col. 8–9	Length of object code in this record in bytes (hexadecimal)
Col. 10–69	Object code, represented in hexadecimal (2 columns per byte of object code)

End record:

Col. 1	E
Col. 2–7	Address of first executable instruction in object program (hexadecimal)

Example Target Program



```
H^COPY  ^001000^00107A
T^001000^1E^141033^482039^001036^281030^301015^482061^3C1003^00102A^0C1039^00102D
T^00101E^15^0C1036^482061^081033^4C0000^454F46^000003^000000
T^002039^1E^041030^001030^E0205D^30203F^D8205D^281030^302057^549039^2C205E^38203F
T^002057^1C^101036^4C0000^F1^001000^041030^E02079^302064^509039^DC2079^2C1036
T^002073^07^382064^4C0000^05
E^001000
```