시스템 프로그래밍 Project #1

- **과제내용**
  - SIC/XE 어셈블러를 개발하기 전에, 프로그램을 입력받을 수 있는 파싱 프로그램 작성
  - 명령어에 해당하는 OPCODE를 찾아서 해당 명령어 옆에 출력
  - 과제에 주어진 C 인터페이스 라이브러리를 사용

- **과제 목적**
  - 입력된 소스코드를 파싱하고 OPCODE를 매핑시킴으로써, 기본적인 SIC/XE 머신을 이해한다.

- **과제 제출 마감**
  - 4/12(화) 수업 시간 전까지
  - 수업 시간 이후부터 당일 오후 6시까지 10 point 패널티 부가
  - 당일 이후부터는 매일 10 point씩 추가 패널티를 부가함

- **제출물**
  - HARD COPY : 출력하여 수업 시간 제출 (프로그램 소스코드 포함)
  - SOFT COPY : 과제 파일 E-campus 제출
  - 레포트 파일은 한글문서, 워드문서로 작성하며, 제목을 제외한 본문의 내용은 **font 10**으로 함

- **레포트 작성 방법 (50 point)**
  - 요구사항
  : 표지 반드시 넣을 것(5p) (학번, 이름, 과제명, 수업 구분<가,나>)
  : 목차 및 내용
  1. 동기/목적(5p) 2. 설계/구현 아이디어(10p) 3. 소스코드(+주석)(10p)
  4. 수행결과(10p) 5. 결론 및 보충할 점(10p)
  - 목차 중 소스코드는 설명해야 할 주요 소스코드(주석포함)만을 작성함
  - 소스코드는 2 column으로 출력할 것 (별도로 아래와 같이 파일로도 제출)
  - 점수 평가에서 레포트의 비중이 높으므로 제출 마감 전까지 성심껏 작성하기 바랍니다.

- **소스코드 제출 방법 (50 point)**
  - 파일 이름: **프로젝트1_이름(학번).zip**
  - 압축파일 내 두개의 디렉토리(**report 와 source**)로 구성
  - 아래 예시와 같이 소스코드를 이클립스 또는 비쥬얼스튜디오의 SP_P1 폴더에 그대로 첨부

※ 주의사항!!!

이클립스 또는 비쥬얼스튜디오 프로젝트 파일을 그대로 첨부 안 할시 소스코드 0점 처리함

- **프로그램 Input/Output은 표시된 Input/Output 문서를 기준으로 함**

- **프로그램 구현에 사용해야할 인터페이스 내용**
  - 매핑을 위한 OPCODE 테이블은 Appendix 참고
  - 아래에 주어진 명세를 참고하여 과제 코드를 구현할 것

    파일명 : my_assembler.h
    내용 : my_assembler.c를 위한 변수선언과 테이블 관리를 위한 구조체 생성

1. 기술되는 구조체 정보

```
// 어셈블리할 소스코드를 토큰 단위로 관리하기 위한 구조체 변수
struct token_unit {
        char *label;                        //명령어 라인 중 label
        char *operator_;                    //명령어 라인 중 operator
        char *operand[MAX_OPERAND];         //명령어 라인 중 operand
        char *comment;                      //명령어 라인 중 comment
};
```

```
// 명령어 목록 파일로부터 정보를 받아와 관리하기 위한 구조체 변수
struct inst_struct {
        char *str;                          // 명령어 이름
        unsigned char op;                   // OPCODE
        int format;                         // 명령어 형식
        int ops;                            // 피연산자 개수
};
```

2. 테이블 관리를 위한 구조체 변수의 배열 선언

```
// 명령어의 정보가 기술된 문자열을 총 256개까지 관리하는 테이블 생성
// SIC/XE 머신의 명령어의 정보를 기술하는 포인터 배열
typedef struct inst_struct inst_struct;
inst_struct *inst[MAX_INST];
int inst_index;

// 어셈블리 할 소스코드를 파일로부터 불러와 라인별로 관리하는 테이블 생성
char *input_data[MAX_LINES];
static int line_num ;

int label_num ;

//  어셈블리할 소스코드를 5000라인까지 관리하는 테이블 생성
typedef struct token_unit token ;
token *token_table[MAX_LINES] ;
```

my_assembler.c 파일에 이미 명시되어있는 함수들을 구현하고, 추가적으로 필요한 함수 구현과 변수 생성은 자유

※ 주의사항(기본 함수는 모두 사용하되, 새롭게 추가한 함수에 대해서는 사용목적을 명시할 것)

*** 입력 파일의 내용**

```
COPY        START   1000        COPY FILE FROM INPUT TO OUTPUT
FIRST       STL     RETADR      SAVE RETURN ADDRESS
CLOOP       JSUB    RDREC       READ INPUT RECORD
            LDA     LENGTH      TEST FOR EOF (LENGTH = 0)
            COMP    ZERO
            JEQ     ENDFIL      EXIT IF EOF FOUND
            JSUB    WRREC       WRITE OUTPUT RECORD
            J       CLOOP       LOOP
ENDFIL      LDA     EOF         INSERT END OF FILE MARKER
            STA     BUFFER
            LDA     THREE       SET LENGTH = 3
            STA     LENGTH
            JSUB    WRREC       WRITE EOF
            LDL     RETADR      GET RETURN ADDRESS
            RSUB                RETURN TO CALLER
EOF         BYTE    C'EOF'
THREE       WORD    3
ZERO        WORD    0
RETADR      RESW    1
LENGTH      RESW    1           LENGTH OF RECORD
BUFFER      RESB    4096        4096-BYTE BUFFER AREA
.
.       SUBROUTINE TO READ RECORD INTO BUFFER
.
RDREC       LDX     ZERO        CLEAR LOOP COUNTER
            LDA     ZERO        CLEAR A TO ZERO
RLOOP       TD      INPUT       TEST INPUT DEVICE
            JEQ     RLOOP       LOOP UNTIL READY
            RD      INPUT       READ CHARACTER INTO REGISTER A
            COMP    ZERO        TEST FOR END OF RECORD (X'00')
            JEQ     EXIT        EXIT LOOP IF EOR
            STCH    BUFFER,X    STORE CHARACTER IN BUFFER
            TIX     MAXLEN      LOOP UNLESS MAX LENGTH
            JLT     RLOOP       HAS BEEN REACHED
EXIT        STX     LENGTH      SAVE RECORD LENGTH
            RSUB                RETURN TO CALLER
INPUT       BYTE    X'F1'       CODE FOR INPUT DEVICE
MAXLEN      WORD    4096
.
.       SUBROUTINE TO WRITE RECORD FROM BUFFER
.
WRREC       LDX     ZERO        CLEAR LOOP COUNTER
WLOOP       TD      OUTPUT      TEST OUTPUT DEVICE
            JEQ     WLOOP       LOOP UNTIL READY
            LDCH    BUFFER,X    GET CHARACTER FROM BUFFER
            WD      OUTPUT      WRITE CHARACTER
            TIX     LENGTH      LOOP UNTIL ALL CHARACTERS
            JLT     WLOOP       HAVE BEEN WRITTEN
            RSUB                RETURN TO CALLER
OUTPUT      BYTE    X'05'       CODE FOR OUTPUT DEVICE
            END     FIRST
```

*** 다음과 같이 출력함** (* '.'으로 시작하는 주석 Line은 output에 출력하지 않아도 됨 - 선택사항)

```
COPY            START   1000
FIRST           STL     RETADR 14
CLOOP           JSUB    RDREC           48
                LDA     LENGTH 00
                COMP    ZERO            28
                JEQ     ENDFIL          30
                JSUB    WRREC           48
                J       CLOOP           3C
ENDFIL          LDA     EOF             00
                STA     BUFFER 0C
                LDA     THREE           00
                STA     LENGTH 0C
                JSUB    WRREC           48
                LDL     RETADR 08
                RSUB                    4C
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            04
                LDA     ZERO            00
RLOOP           TD      INPUT           E0
                JEQ     RLOOP           30
                RD      INPUT           D8
                COMP    ZERO            28
                JEQ     EXIT            30
                STCH    BUFFER,X        54
                TIX     MAXLEN          2C
                JLT     RLOOP           38
EXIT            STX     LENGTH 10
                RSUB                    4C
INPUT           BYTE    X'F1'
MAXLEN          WORD    4096
.
.       SUBROUTINE TO WRITE RECORD FROM BUFFER
.
WRREC           LDX     ZERO            04
WLOOP           TD      OUTPUT          E0
                JEQ     WLOOP           30
                LDCH    BUFFER,X        50
                WD      OUTPUT          DC
                TIX     LENGTH 2C
                JLT     WLOOP           38
                RSUB                    4C
OUTPUT          BYTE    X'05'
                END     FIRST
```