System Software: Onsite Test 2 (An implementation of Macro Processor)

Description: Implementation of a one-pass macro processor for SIC/XE machine

- Input: an SIC assembly program with macro (no label in it)
- Output: an expanded assembly program

Requirement:

- Packed your project as a zip file which includes your source code, input SIC, expanded program, output(DEFTAB, NAME TAB, ARGTAB) and a Readme
- The test procedure should be described clearly in the Readme file.
- The file name of your Zip file should include your school id number.

Grading

- Basic requirements (100%)
  1. The basic Macro processor functions
     - Two pass macro processor
       - Pass 1: All macro definitions are processed.
       - Pass 2: All macro invocation statements are expanded.
       - Definition table (DEFTAB): Macro definitions and body are stored.
         - Comment lines are skipped.
         - References to macro instruction parameters are converted to a positional notation.
       - Name table (NAMTAB): Macro names with pointers to the beginning and end of the macro in DEFTAB.
       - Argument table (ARGTAB): Store invocation parameters that are used during the expansion of macro invocation.
- (25%) Machine-independent macro processor features (5 points for each feature, 25 points is the maximum)
  2. One pass algorithm
  3. Label in Macro definition
  4. conditional expansion (If, While)
  5. keyword parameter
  6. Concatenation of Macro parameters

One-Pass Macro Algorithm (Cont.)

```
begin {macro processor}
    EXPANDING := FALSE
    while OPCODE ≠ 'END' do
        begin
            GETLINE
            PROCESSLINE
        end {while}
end {macro processor}
```

```
procedure PROCESSLINE
    begin
        search NAMTAB for OPCODE
        if found then
            EXPAND
        else if OPCODE = 'MACRO' then
            DEFINE
        else write source line to expanded file
    end {PROCESSLINE}
```

```
procedure DEFINE
    begin
        enter macro name into NAMTAB
        enter macro prototype into DEFTAB
        LEVEL := 1
        while LEVEL > 0 do
            begin
                GETLINE
                if this is not a comment line then
                    begin
                        substitute positional notation for parameters
                        enter line into DEFTAB
                        if OPCODE = 'MACRO' then
                            LEVEL := LEVEL + 1
                        else if OPCODE = 'MEND' then
                            LEVEL := LEVEL - 1
                    end {if not comment}
            end {while}
        store in NAMTAB pointers to beginning and end of definition
    end {DEFINE}
```

```
procedure EXPAND
    begin
        EXPANDING := TRUE
        get first line of macro definition (prototype) from DEFTAB
        set up arguments from macro invocation in ARGTAB
        write macro invocation to expanded file as a comment
        while not end of macro definition do
            begin
                GETLINE
                PROCESSLINE
            end {while}
        EXPANDING := FALSE
    end {EXPAND}
```

```
procedure GETLINE
    begin
        if EXPANDING then
            begin
                get next line of macro definition from DEFTAB
                substitute arguments from ARGTAB for positional notation
            end {if}
        else
            read next line from input file
    end {GETLINE}
```

```
 5        COPY       START    0                  COPY FILE FROM INPUT TO OUTPUT
10        RDBUFF     MACRO    &INDEV,&BUFADR,&RECLTH
15        .
20        .          MACRO TO READ RECORD INTO BUFFER
25        .
30                   CLEAR    X                  CLEAR LOOP COUNTER
35                   CLEAR    A
40                   CLEAR    S
45                  +LDT      #4096              SET MAXIMUM RECORD LENGTH
50                   TD       =X'&INDEV'         TEST INPUT DEVICE
55                   JEQ      *-3                LOOP UNTIL READY
60                   RD       =X'&INDEV'         READ CHARACTER INTO REG A
65                   COMPR    A,S                TEST FOR END OF RECORD
70                   JEQ      *+11               EXIT LOOP IF EOR
75                   STCH     &BUFADR,X          STORE CHARACTER IN BUFFER
80                   TIXR     T                  LOOP UNLESS MAXIMUM LENGTH
85                   JLT      *-19                 HAS BEEN REACHED
90                   STX      &RECLTH            SAVE RECORD LENGTH
95                   MEND
100       WRBUFF     MACRO    &OUTDEV,&BUFADR,&RECLTH
105       .
110       .          MACRO TO WRITE RECORD FROM BUFFER
115       .
120                  CLEAR    X                  CLEAR LOOP COUNTER
125                  LDT      &RECLTH
130                  LDCH     &BUFADR,X          GET CHARACTER FROM BUFFER
135                  TD       =X'&OUTDEV'        TEST OUTPUT DEVICE
140                  JEQ      *-3                LOOP UNTIL READY
145                  WD       =X'&OUTDEV'        WRITE CHARACTER
150                  TIXR     T                  LOOP UNTIL ALL CHARACTERS
155                  JLT      *-14                 HAVE BEEN WRITTEN
160                  MEND
165       .
170       .          MAIN PROGRAM
175       .
180       FIRST      STL      RETADR             SAVE RETURN ADDRESS
190       CLOOP      RDBUFF   F1,BUFFER,LENGTH   READ RECORD INTO BUFFER
195                  LDA      LENGTH             TEST FOR END OF FILE
200                  COMP     #0
205                  JEQ      ENDFIL             EXIT IF EOF FOUND
210                  WRBUFF   05,BUFFER,LENGTH   WRITE OUTPUT RECORD
215                  J        CLOOP              LOOP
220       ENDFIL     WRBUFF   05,EOF,THREE       INSERT EOF MARKER
225                  J        @RETADR
230       EOF        BYTE     C'EOF'
235       THREE      WORD     3
240       RETADR     RESW     1
245       LENGTH     RESW     1                  LENGTH OF RECORD
250       BUFFER     RESB     4096               4096-BYTE BUFFER AREA
```