Assembler Pseudocode.

2 pass assembler for SIC/XE

Pass 1:

```
BEGIN
  initialize Scnt, Locctr, ENDval, and Errorflag to 0
  WHILE Sourceline[Scnt] is a comment
    BEGIN
      increment Scnt
    END {while}
  Breakup Sourceline[Scnt]
  IF Opcode = 'START' THEN
    BEGIN
      convert Operand from hex and save in Locctr and ENDval
      IF Label not NULL THEN
        Insert (Label, Locctr) into Symtab
      ENDIF
      increment Scnt
      Breakup Sourceline[Scnt]
    END
  ENDIF
  WHILE Opcode <> 'END'
    BEGIN
      IF Sourceline[Scnt] is not a comment THEN
        BEGIN
          IF Label not NULL THEN
            Xsearch Symtab for Label
            IF not found
              Insert (Label, Locctr) into Symtab
            ELSE
              set errors flag in Errors[Scnt]
            ENDIF
          ENDIF
          Xsearch Opcodetab for Opcode
          IF found THEN
            DO CASE
              1. Opcode is 'RESW' or 'RESB'
                BEGIN
                  increment Locctr by Storageincr
                  IF error THEN
                    set errors flag in Errors[Scnt]
                  ENDIF
                END {case 1 (RESW or RESB)}
              2. Opcode is 'WORD' or 'BYTE' THEN
                BEGIN
                  increment Locctr by Storageincr
                  IF error THEN
                    set errors flag in Errors[Scnt]
                  ENDIF
                END {case 2 (WORD or BYTE)}
              3. OTHERWISE
```

```
                BEGIN
                  increment Locctr by Opcodeincr
                  IF error THEN
                    set errors flag in Errors[Scnt]
                  ENDIF {case 3 (default)}
                END
            ENDCASE
          ELSE
            /* directives such as BASE handled here or */
            set errors flag in Errors[Scnt]
          ENDIF
        END {IF block}
      ENDIF
    increment Scnt
    Breakup Sourceline[Scnt]
  END {while}
  IF Label not NULL THEN
    Xsearch Symtab for Label
    IF not found
      Insert (Label, Locctr) into Symtab
    ELSE
      set errors flag in Errors[Scnt]
    ENDIF
  ENDIF
  IF Operand not NULL
    Xsearch Symtab for Operand
    IF found
      install in ENDval
    ENDIF
  ENDIF
END {of Pass 1}
```

Pass 2:

```
BEGIN
  initialize Scnt, Locctr, Skip, and Errorflag to 0
  write assembler report headings
  WHILE Sourceline[Scnt] is a comment
    BEGIN
      append to assembler report
      increment Scnt
    END {while}
  Breakup Sourceline[Scnt]
  IF Opcode = 'START' THEN
    BEGIN
      convert Operand from hex and save in Locctr
      append to assembler report
      increment Scnt
      Breakup Sourceline[Scnt]
    END
  ENDIF
  format and place the load point on object code array
  format and place ENDval on object code array, index ENDloc
```

```
WHILE Opcode <> 'END'
  BEGIN
    IF Sourceline[Scnt] is not a comment THEN
      BEGIN
        Xsearch Opcodetab for Opcode
        IF found THEN
          DO CASE
            1. Opcode is 'RESW' or 'RESB'
              BEGIN
                increment Locctr by Storageincr
                place '!' on object code array
                replace the value at index ENDloc with loader address
                format and place Locctr on object code array
                format and place ENDval on object code array, index ENDloc
                set Skip to 1
              END
            2. Opcode is 'WORD' or 'BYTE'
              BEGIN
                increment Locctr by Storageincr
                Dostorage to get Objline
                IF error THEN
                  set errors flag in Errors[Scnt]
                ENDIF
              END
            3. OTHERWISE
              BEGIN
                increment Locctr by Opcodeincr
                Doinstruct to get Objline
                IF error THEN
                  set errors flag in Errors[Scnt]
                ENDIF
              END
          ENDCASE
        ELSE
          /* directives such as BASE handled here or */
          set errors flag in Errors[Scnt]
        ENDIF
      END
    ENDIF
    append to assembler report
    IF Errors[Scnt] <> 0 THEN
      BEGIN
        set Errorflag to 1
        append error report to assembler report
      END
    ENDIF
    IF Errorflag = 0 and Skip = 0 THEN
      BEGIN
        place Objline on object code array
      END
    ENDIF
    IF Skip = 1 THEN
      set Skip to 0
    ENDIF
```

```
      increment Scnt
      Breakup Sourceline[Scnt]
    END {while}
  place '!' on object code array
  IF Errorflag = 0 THEN
    transfer object code array to file
  ENDIF
END {of Pass 2}
```