Q1. Program to set ALPHA= GAMMA*BETA-9 using Register Operation.

—

```
        LDA     GAMMA
        MUL     BETA
        SUB     #9
        STA     ALPHA
ALPHA   RESW    1
BETA    RESW    1
GAMMA   RESW    1
```

Q2. SIC/XE program to copy a character string to another string.

—

```
        LDX     #0
        LDS     #10
LOOP    LDCH    STR1,X
        STCH    STR2,X
        TIXR    S
        JLT     LOOP
STR1    RESB    10
STR2    RESB    10
```

Q3. SIC/XE program to write a subroutine to read a record into a buffer.

—

```
        JSUB    RDREC
RDREC   LDX     #0
        LDT     #100
        LDS     #0
RLOOP   TD      INDEV
        JEQ     RLOOP
```

```
            RD       INDEV
            COMPR    A,S
            JEQ      EXIT
            STCH     BUFFER,X
            TIXR     T
            JLT      RLOOP
EXIT        STX      LENGTH
            RSUB
INDEV       BYTE     X'F1'
LENGTH      RESW     1
BUFFER      RESB     100
```

Q4. SIC/XE program to find the minimum element in the array and store result in MIN.

—

```
            LDS      #3
            LDT      #60
            LDX      #0
LOOP1       LDA      ALPHA,X
            COMP     MIN
            JGT      LOOP2
            STA      MIN
LOOP2       ADDR     S,X
            COMP     X,T
            JLT      LOOP1

ALPHA       RESW     20
MIN         WORD     1000
```

Q5. SIC/XE program to multiply two arrays.

—

```
        LDS     #3
        LDT     #60
        LDX     #0
LOOP    LDA     ALPHA,X
        MUL     BETA,X
        STA     GIAMMA,X
        ADDR    S,X
        COMP    X,T
        JLT     LOOP

ALPHA   RESW    20
BETA    RESW    20
GIAMMA  RESW    20
```

Q6. Suppose that ALPHA is an array of 100 words. Write a sequence of instruction for SIC/XE to arrange the 100 words in ascending order and store result in an array BETA of 100 elements.

—

```
SORT    START   0
OUTER   LDX     INDEX
~~LDS     ARR1,X~~
        LDS     ARR1,X
        LDX     #0
INNER   LDT     ARR1,X
        COMPR   S,T
        JLT     LOOP
        JEQ     LOOP
```

```
          RMO     S,A
          RMO     T,S
          RMO     A,T
          RMO     X,A
          LDX     INDEX
          STS     ARR1,X
          RMO     A,X
          STT     ARR1,X
LOOP      RMO     X,A
          ADD     #3
          COMP    LENGTH
          RMO     A,X
          JLT     INNER
          LDA     INDEX
          ADD     #3
          COMP    LENGTH
          STA     INDEX
          JLT     OUTER
          ~~ARR1~~    ~~R~~
ARR1      RESW    10
LENGTH    WORD    30
INDEX     WORD    0
```

Q7. Program to clear 20 byte string to empty.

—

SIC

```
        LDX     INDEX
        LDCH    BLANK
LOOP    STCH    STR1,X
        TIX     TWENTY
        JLT     LOOP
INDEX   WORD    0
BLANK   BYTE    C' '
STR1    RESB    20
TWENTY  WORD    20
```

SIC/XE

```
        LDX     #0
        LDS     #20
        LDCH    #0
LOOP    STCH    STR1,X
        TIXR    S
        JLT     LOOP
STR1    RESB    20
```

Q8. Write sequence of instructions for SIC/XE to divide BETA by GAMMA setting ALPHA to the integer portion of quotient and DELTA to the remainder.

—

```
        LDA     BETA
        LDS     GAMMA
        DIVR    S,A
        STA     ALPHA
        MULR    S,A
        LDS     BETA
        SUBR    A,S
        STS     DELTA
BETA    RESW    1
GAMMA   RESW    1
ALPHA   RESW    1
DELTA   RESW    1
```

Q9. Write SIC instructions to swap the values of ALPHA and BETA.

—

```
          LDA     ALPHA
          STA     GAMMA
          LDA     BETA
          STA     ALPHA
          LDA     GAMMA
          STA     BETA

ALPHA     RESW    1
BETA      RESW    1
GAMMA     RESW    1
```

Q10. Write a sequence of instructions in SIC/XE to add 2 arrays of 100 integers.

—

```
          LDS     #3
          LDT     #300
          LDX     #0
LOOP      LDA     ALPHA,X
          ADD     BETA,X
          STA     GAMMA,X
          ADDR    S,X
          COMPR   X,T
          JLT     LOOP

ALPHA     RESW    100
BETA      RESW    100
GAMMA     RESW    100
```