CST-305

SYSTEM SOFTWARE

Assignment-1

Name: Sarath Sasi
Branch: S5, CSE
Roll no: 58
College of Engg: Adoor.

#1 Write a sequence of instructions for SIC to ALPHA equal to the product of BETA and GAMMA.

Soln:

```
LDA   BETA.
MUL   GAMMA.
STA   ALPHA.
  :
ALPHA  RESW 1
BETA   RESW 1
GAMMA  RESW 1
```

#2. Write a sequence of instructions for SIC/XE to set ALPHA equal to 4 * BETA − 9. Use immediate addressing for the constants.

Soln:

```
LDA   BETA
LDS   #4
MULR  S,A
SUB   #9
STA   ALPHA
  :
BETA   RESW 1
ALPHA  RESW 1
```

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

Soln:

```
LDA   ALPHA
STA   GAMMA
LDA   BETA
STA   ALPHA
LDA   GAMMA
STA   BETA
  :
ALPHA  RESW 1
BETA   RESW 1
GAMMA  RESW 1
```

#4. Write a sequence of instructions for SIC/XE to divide BETA by GAMMA, setting ALPHA to integer portion of the quotient and DELTA to the remainder. Use register-to-register instructions to make the calculation as efficient as possible.

Soln:

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

#5. Write a sequence of instructions for SIC/XE to clear a 20-byte string to all blanks. Use immediate addressing and register-to-register instructions to make the process as efficient as possible.

Soln:

```
        LDT   #20
        LDX   #0
LOOP    LDCH  #0
        STCH  STR1, X
        TIXR  T
        JLT   LOOP
        .
STR1    RESW  20
```

#6 Suppose that ALPHA and BETA are the two arrays of 100 words. Another array of GAMMA elements are obtained by multiplying the corresponding ALPHA element by 4 and adding the corresponding BETA elements.

Soln:

```
        LDS   #3
        LDT   #300
        LDX   #0
ADDLOOP LDA   ALPHA, X
        MUL   #4
        ADD   BETA, X
        STA   GAMMA, X
        ADDR  S, X
        COMPR X, T
        JLT   ADDLOOP
        :
        :
ALPHA   RESW  100
BETA    RESW  100
GAMMA   RESW  100
```

#7. Suppose that ALPHA is an array of 100 words. Write a sequence of instructions for SIC/XE to set all 100 elements of the array to 0. Use immediate addressing and register-to-register instructions to make the process as efficient as possible.

Soln:

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

```
        JLT   LOOP
        :
ALPHA   RESW  100
```

#8. Suppose that ALPHA is an array of 100 words. Write a sequence of instructions for SIC/XE to find the maximum element in the array and store results in MAX

Soln:

```
        LDS   #3
        LDT   #300
        LDX   #0
CLOOP   LDA   ALPHA,X
        COMP  MAX
        JLT   NOCH
        STA   MAX
NOCH    ADDR  S,X
        COMPR X,T
        JLT   CLOOP
        :
ALPHA   RESW  100
MAX     WORD  -32768
```

#9. Suppose that RECORD contains a 100-byte record. Write a sub.routine for SIC that will write this record on to device 05.

Soln:

```
        JSUB  WRREC

WRREC   LDX   ZERO
WLOOP   TD    OUTPUT
        JEQ   WLOOP
```

```
        LDCH    RECORD, X
        WD      OUTPUT
        TIX     LENGTH
        JLT     WLOOP
        RSUB
        :
ZERO    WORD    0
LENGTH  WORD    1
OUTPUT  BYTE x "05"
RECORD  RESB    100
```

#10. Write a subroutine for SIC/XE that will read a record into a buffer. The record may be any length from 1 to 100 bytes. The end of record is marked with a "null" character (ASCII code 00). The subroutine should place the length of the record read into a variable named LENGTH. Use immediate addressing and register-to-register instructions to make the process as efficient as possible.

Soln:

```
        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
```

```
EXIR    STX     LENGTH
        RSUB
        :
INDEV   BYTE    X'F1'
LENGTH  RESW    1
BUFFER  RESB    100
```