

Systems Programming assignment 1

① SIC

LDA	BETA	; A = Beta
MUL	GAMMA	; A = A * GAMMA
STA	ALPHA	; ALPHA = A = BETA * GAMMA
:		

ALPHA	RESW	1	; 1 word variable
BETA	RESW	1	
GAMMA	RESW	1	

② SIC

LDA	BETA	; A = BETA
DIV	GAMMA	; A = A / GAMMA
STA	ALPHA	; ALPHA = A = BETA / GAMMA
:		

ALPHA	RESW	1
BETA	RESW	1
GAMMA	RESW	1

(3) ^{SIC}

	LDA	ZERO	; initialize index value to 0
	STA	INDEX	; A = 0
Zloop	LDX	INDEX	; load index value into Reg X
	LDA	ZERO	; load A = 0
	STA	ALPHA, X	; Store the zero into ALPHA directly
	LDA	INDEX	; load the index to increment it in the next step
	ADD	THREE	; increment value of index by 3 as 1 word = 3 bytes
	STA	INDEX	; store the next index
	CMP	K 100	; Compare the new index to 100
	JLT	Zloop	; loop if index is less than 100
	:		

INDEX	RESW	1	; 1 word variable
ALPHA	RESW	100	
ZERO	WORD	0	; 1 word constant '0'
THREE	WORD	3	
K 100	WORD	100	

④ SIC

JSUB WRITE

; jump to write subroutine (L=PC
; PC = ~~PC~~ + 1)

WRITE LDX ZERO

; load index value '0' into RegX

WLOOP TD OUTDEV

; test output device

TEQ WLOOP

; loop until device is ready

LDRH RECORD, X

; load ^{from} record into reg A directly

WD OUTDEV

; write 1 byte to output device

TIX K100

; increment & compare it to 100

JLT WLOOP

; loop if less than 100

RSUB

; PC = L

OUTDEV BYTE X'05'

; 1 byte constant

RECORD RESB 100

; 100 bytes variable

ZERO WORD 0

K100 WORD 100

5^{SEC}

FOREV CLEAR A
 STA ofs ; clears the cache

LOOP LDA ofs ; start of print loop
 ADD ONE

W1 TD OUTDEV ; write to output device
 JEQ W1
 WD OUTDEV

 LDA ofs # ; increase offset
 ADD ONE
 STA ofs
 COMP COUNT
 JLT LOOP
 .
 .

OUTDEV	BYTE	X'01'
ONE	WORD	1
COUNT	WORD	10 10
ofs	RESW	1
	RESW	
ofs	RESW	1