

Assignment 5

Yiting Long 301183087

1.

```
RAMStart EQU $0000 ; absolute address to place variables
RAMEnd EQU $0BFF ; last address of RAM, used to init stack pointer
ROMStart EQU $0A00 ; absolute address to place code/constant data

; variable/data section

ORG RAMStart
; Insert here your data definition.
ORG #080A
firstNum DC.B 199 ;define label firstNum and initialize to value decimal 199 and store at location 080A
ORG #081B
secondNum DC.B 87 ;define label secondNum and initialize to value decimal 87 and store at location 081B
ORG #082E
thirdNum DC.B 11 ;define label thridNum and initialize to value decimal 11 and store at location 082E
ORG #0811
result DS.B 1 ;reserve 1 bit of memory for the result and store at location 0811

; code section
ORG ROMStart

Entry:
LDAA firstNum ;load value at label firstNum into accumulator A

SUBA secondNum ;subtract value at label secondNum with accumulator A and
;store result back in A

ADDA thirdNum ;add value at label thirdNum with accumulator A and
;store result back in A

STAA result ;store accumulator A into address result

STOP ;end of program
```

2.

```

RAMStart    EQU    $0000            ; absolute address to place variables
RAMEnd      EQU    $0BFF            ; last address of RAM, used to init stack pointer
ROMStart    EQU    $0A00            ; absolute address to place code/constant data

; variable/data section

        ORG RAMStart
; program that calls a subroutine to take
; the bitwise XNOR out of two parameters
        ORG $0800
valueX DC.B #3A
valueY DC.B #18
RES     DC.W 0 ;the final answer

;initialize SP, A, B, X, Y
        ORG #0A00
ENTRY    LDS #00FF ;load stack pointer with an address
        LDD #0      ;reset both A and B to zero
        LDX #0
        LDY #0

        ;pass parameters and setup room for the return value for subroutine
        PSHX        ;set up 2 byte space on stack for the return value

        LDAA valueX;parameter X for function
        LDAB valueY;parameter Y for function
        PSHA        ;push parameter X onto the stack
        PSHB        ;push parameter Y onto the stack

        ;call the subroutine
        JSR XNOR     ;jump to the subroutine

        ;clean stack and get return value
        INS          ;increment stack pointer twice
        INS          ;because there are two parameters in the stack

        PULX        ;pop return value
        STX RES      ;store result

        STOP

XNOR     PSHA        ;save all registers
        PSHB

```

```

        PSHX
        PSHY
        TSX          ;transfer stack pointer

        LDAA 9,X      ;load parameter into A
        EORA 8,X      ;xor parameter 1 and parameter 2
                     ;result store back into A
        COMA          ;takes 1's complement of A
        STAA 10,X     ;store return value on stack in byte
                     ;reserved for it

        PULY
        PULX
        PULB
        PULA          ;reset all register

        RTS          ;return from subroutine

```

```

RAMStart EQU $0000 ; absolute address to place variables
RAMEnd EQU $0BFF ; last address of RAM, used to init stack pointer
ROMStart EQU $0A00 ; absolute address to place code/constant data

; variable/data section

ORG RAMStart
; program that calls a subroutine to take
; the bitwise XNOR out of two parameters
ORG $0800
RES DC.W 0 ;the final answer

;initialize SP, A, B, X, Y
ORG $0A00

ENTRY LDS #00FF ;load stack pointer with an address
LDD #0 ;reset both A and B to zero
LDX #0
LDY #0

;pass parameters and setup room for the return value for subroutine
PSHX ;set up 2 byte space on stack for the return value

LDAA 5 ;parameter 5 for function
PSHA ;push parameter X onto the stack

;call the subroutine
JSR EIGHT ;jump to the subroutine

;clean stack and get return value
INS ;increment stack pointer

PULX ;pop return value
STX RES ;store result

STOP

EIGHT PSHA ;save all registers
PSHB
PSHX
PSHY
TSX ;transfer stack pointer

LDAB 8,X ;load parameter into B
LDAA 5 ;load parameter 5 into A

```

```

ABA                ;add B to A and store the result back into A
ABA                ;add 5 to A 7 times so the result equals
ABA                ;to 5 * 8
ABA
ABA
ABA
ABA
STAA 9,X           ;store return value on stack in byte
                   ;reserved for it
PULY
PULX
PULB
PULA               ;reset all register
RTS                ;return from subroutine

```