پاسخ تمرین سری دوم

سوال 1 )

```
.include "m64def.inc"

.org 0x0000
    JMP main

.org 0x0050
main:
    LDI R16, 0x00
    OUT DDRA, R16       ; تنظیم PORTA به عنوان ورودی
    OUT DDRB, R16       ; تنظیم PORTB به عنوان ورودی

    LDI R16,0x0FF
    OUT DDRC, R16       ; تنظیم PORTC به عنوان خروجی
    OUT DDRD, R16       ; تنظیم PORTD به عنوان خروجی

    LDI R26, 0x00       ; X pointer
    LDI R27, 0x20

Loop:
    IN R17,PINA         ; خواندن مقدار از PORTA
    IN R18,PINB         ; خواندن مقدار از PORTB

    CP R18, R17         ; مقایسه دو مقدار
    BRCS B_less         ; اگر B < A
    BREQ loop
    MOV R20, R17        ; R20 = مقدار کوچکتر
    MOV R21, R18        ; R21 = مقدار بزرگتر
    JMP next

B_less:
    MOV R20, R18        ; R20 = مقدار کوچکتر
    MOV R21, R17        ; R21 = مقدار بزرگتر
next:
    MOV R22, R20
    ROR R22             ; بررسی زوج یا فرد بودن عدد کوچکتر
    BRCS Lessodd        ; اگر فرد بود
    JMP CheckBig

Lessodd:
    LDI R16, 7
    MUL R20, R16        ; ضرب عدد کوچکتر در 7
    OUT PortC, R0
    OUT PortD, R1

CheckBig:
    MOV R22, R21
    ROR R22             ; بررسی زوج بودن عدد بزرگتر
    BRCC BigEven        ; اگر زوج بود
    JMP Loop

BigEven:
    CP R21, 27
    BRCS end            ; اگر کمتر از 27 شد، پایان
    SUBI R21, 27
    JMP BigEven

end:
    ST X+,R21
    JMP Loop
```

سوال۲ )

```
.include "m64def.inc"
.org 0x0000
    JMP main
.org 0x0050
main:
    LDI R26, 0x00
    LDI R27, 0x40
    LDI R28, 0x00
    LDI R29, 0x60

Loop:
    LD R16, X+

    CPI R16, 2
    BRLO next

    CPI R16, 2
    BREQ prime

    MOV R17, R16
    ANDI R17, 1
    BREQ next

    MOV R17, R16
    LSR R17

check_prime:
    CPI R17, 1
    BREQ prime

    MOV R18, R16

mod_loop:
    CP R18, R17
    BRLO mod_done
    SUB R18, R17
    JMP mod_loop

mod_done:
    CPI R18, 0
    BREQ next
    DEC R17
    JMP check_prime

prime:
    LDI R17, 1

sqrt_loop:
    MOV R18, R17
    MUL R18, R18
    MOV R19, R0
    CLR R1
    CP R19, R16
    BRSH sqrt_done
    INC R17
    JMP sqrt_loop

sqrt_done:
    DEC R17
    ST Y+, R17

next:
    CPI R26, 0x00
    BRNE Loop

    CPI R27, 0x50
    BRNE Loop

end:
    JMP end
```

سوال ۳ )

```
#include "M64DEF.INC"
org 0x0000
jmp main
main:
    LDI R16, 0xFF
    OUT DDRR, R16

    ; محاسبه B9 + 5C
    LDI R17, 0xB9
    LDI R18, 0x5C
    ADD R17, R18

    ; حتما بزرگتر <- Carry=1 اگر
    BRCS do_not

    LDS R19, 0x049F

    ; مقایسه انجام شود <- Carry=0 اگر
    CP R17, R19
    BRGT do_not
    JMP else_part

    ; اگر بزرگتر بود
do_not:
    IN R22, PINF
    COM R22
    OUT PORTF, R22
    JMP end

else_part:
    LDS R23, 0x3000

    ; گرفتن 4 بیت پایین
    ANDI R23, 0x0F

    ; شیفت چپ
    LSL R23
    LSL R23
    LSL R23
    LSL R23 ; معادل جابه‌جایی 4 بیت به بالا (x16)
    LDS R24, 0x4000

    ; پاک کردن 4 بیت بالا
    ANDI R24, 0x0F

    OR R24, R23
    STS 0X4000, R24
end: JMP end
```

سوال 4)

```
.include "M64DEF.INC"
.org 0x0000
jmp main

main:
    LDI R26, 0x00
    LDI R27, 0x20
    LDI R28, 0x00
    LDI R29, 0x10
    LDI R30, 0x00
    LDI R31, 0x30

    CLR R25
    CLR R24
    CLR R18 ; فلگ برای قسمت پ
    LDI R19, 1

Loop:
    CLR R18

    LD R16, X+

    ; بررسی بازه ها
    CPI R16, 91
    BRLO check_second
    CPI R16, 120
    BRSH check_second

    LDI R18, 1
    JMP range_ok

check_second:
    CPI R16, 31
    BRLO check4
    CPI R16, 60
    BRSH check4

    LDI R18, 1

range_ok:
    ST Y+, R16

check4:
    ; بررسی مضرب 4
    MOV R17, R16
    ANDI R17, 0x03
    BRNE next

    ST Z+, R16

    INC R18

    CPI R18, 2
    BREQ both
    JMP next

both:
    ADD R25, R19
    BRCC noCarry
    JMP withCarry

noCarry:
    STS 0x4DEE, R25
    JMP next

withCarry:
    INC R24
    STS 0x4DEE, R25
    STS 0x4DEF, R24

next:
    CPI R26, 0x00
    BRNE Loop
    CPI R27, 0x21
    BRNE Loop

end: JMP end
```