

Assignment 4

1.

```
;average

MOV R0, #20h ; base address of array
MOV R1, #10h ; array length

loop:
    ADD A, @R0
    INC R0
    DJNZ R1, loop

MOV B, #10h

DIV AB
MOV 30h, A

stop:
SJMP stop;
END
```

2.

```
MOV R0, #0
MOV R1, #0

loop:
    MOV A, P1
    JZ stop

    ANL A, #01H
    ADD R0, A

    MOV A, R1
    INC A
    MOV R1, A

    RRC P1
    JNC loop

stop:
    NOP
    SJMP stop
```

3.

```
ORG 0x00

START:
    MOV SP, #0x7F

    MOV R0, #0
    MOV R1, #7Fh

    ; Load the sequence of numbers into memory
    MOV R2, #LENGTH
    MOV DPTR, #SEQUENCE
    MOVX A, @DPTR
    PUSH ACC
    INC DPTR
LOOP_LOAD:
    MOVX A, @DPTR
    PUSH ACC
    INC DPTR
    DJNZ R2, LOOP_LOAD
```

```

; Check if the sequence is a palindrome
MOV R0, #0
MOV R2, #LENGTH
DEC R2
MOV DPTR, #SEQUENCE
MOV A, R2
ADD A, R0
JC PALINDROME
INC R0
LOOP_COMPARE:
    MOVX A, @DPTR
    INC DPTR
    MOV R3, A
    MOVX A, @R1
    CMP A, R3
    JNZ NOT_PALINDROME
    INC R0
    DJNZ R2, LOOP_COMPARE
    JMP PALINDROME

NOT_PALINDROME:
    MOV A, #0
    SJMP END

PALINDROME:
    MOV A, #1

END:
    NOP
    SJMP $

LENGTH EQU 8
SEQUENCE DB 1, 2, 3, 4, 4, 3, 2, 1

```

4.

a.

```

ORG 0x00

START:
    MOV A, #0xFF
    MOV P1, A

    MOV A, #0x0A
    MOV B, #0x05

    ADD A, B

    MOV P1, A

END:
    NOP
    SJMP $

```

b.

```
ORG 0x00

START:
    MOV A, #0xFF
    MOV P1, A
    MOV P2, A

    MOV R0, #0x0A
    MOV R1, #0x00
    MOV R2, #0x05
    MOV R3, #0x00

    ; Add the least significant bytes first
    ADD A, R0
    MOV P1, A

    MOV A, R1
    ADDC A, R3
    MOV P2, A

    ; Add the most significant bytes
    MOV A, R0
    ADD A, R2
    MOV P1, A

    MOV A, R1
    ADDC A, R3
    MOV P2, A

END:
    NOP
    SJMP $
```

c.

```

ORG 0x00

START:
    MOV A, #0xFF
    MOV P1, A
    MOV P2, A
    MOV A, #0x00
    MOV P3, A

    ; Read first 8-bit number from port P1
    MOV A, P1
    MOV R0, A

    ; Read second 8-bit number from port P2
    MOV A, P2
    ADD A, R0
    MOV P3, A

END:
    NOP
    SJMP $

```

d.

```

ORG 0x00

START:
    MOV A, #0xFF
    MOV P1, A
    MOV P2, A
    MOV P3, A
    MOV P4, A

    ; MSB the first 16-bit number from port P1
    MOV A, P1
    MOV R0, A

    ; LSB the first 16-bit number from port P2
    MOV A, P2
    ADD A, R0
    MOV R1, A

    ; MSB the second 16-bit number from port P3
    MOV A, P3
    ADD A, R1
    MOV P3, A

    ; LSB the second 16-bit number from port P4
    MOV A, P4
    ADDC A, #0x00
    MOV P4, A

END:
    NOP
    SJMP $

```

