Q1:

```
;--------------------------------------------
; Program: Add two arrays (5 bytes each)
;--------------------------------------------
        LIST P=18F4520
        ORG 0x00

ARRAY1  EQU 0x20        ; starting address of ARRAY1
ARRAY2  EQU 0x25        ; starting address of ARRAY2
RESULT  EQU 0x30        ; starting address of RESULT
COUNT   EQU 0x35        ; loop counter

        MOVLW 0x05
        MOVWF COUNT

        LFSR 0, ARRAY1   ; FSR0 -> ARRAY1
        LFSR 1, ARRAY2   ; FSR1 -> ARRAY2
        LFSR 2, RESULT   ; FSR2 -> RESULT

ADD_LOOP:
        MOVF POSTINC0, W ; W = *FSR0++
        ADDWF POSTINC1, W ; W = W + *FSR1++
        MOVWF POSTINC2    ; *FSR2++ = W

        DECFSZ COUNT, F
        GOTO ADD_LOOP

        END
```

Q2:

```
;--------------------------------------------
; Program: Binary to ASCII (0-9)
;--------------------------------------------
        LIST P=18F4520
        ORG 0x00

NUM     EQU 0x20
CHAR    EQU 0x21

        MOVF NUM, W      ; Get number (0-9)
        ADDLW 0x30       ; Convert to ASCII ('0' = 0x30)
        MOVWF CHAR       ; Store result

        END
```

Q3:

```
;--------------------------------------------
; Program: Toggle case of ASCII letter
;--------------------------------------------
```

```
        LIST P=18F4520
        ORG 0x00

CHAR    EQU 0x20

        MOVF CHAR, W
        MOVWF TEMP

        MOVLW 'A'
        CPFSLT TEMP       ; if CHAR < 'A' skip
        GOTO CHECK_LOWCASE
        MOVLW 'Z'
        CPFSGT TEMP       ; if CHAR > 'Z' skip
        GOTO TO_LOWER
        GOTO CHECK_LOWCASE

TO_LOWER:
        MOVLW 0x20        ; difference between 'A' and 'a'
        ADDWF CHAR, F
        GOTO DONE

CHECK_LOWCASE:
        MOVLW 'a'
        CPFSLT TEMP
        GOTO DONE
        MOVLW 'z'
        CPFSGT TEMP
        GOTO DONE

TO_UPPER:
        MOVLW 0x20
        SUBWF CHAR, F

DONE:
```

END

Q5:

```
;--------------------------------------------
; Program: Student Marks Processing
;--------------------------------------------
        LIST P=18F4520
        ORG 0x00

MARKS   EQU 0x20       ; 5 bytes: marks of 5 students
AVG     EQU 0x25
COUNT50 EQU 0x26
SUM     EQU 0x27
TEMP    EQU 0x28
N       EQU 0x29

        CLRF SUM
        CLRF COUNT50

        MOVLW 0x05
        MOVWF N

        LFSR 0, MARKS
```

```
MARK_LOOP:
        MOVF POSTINC0, W
        ADDWF SUM, F           ; SUM += mark

        MOVWF TEMP
        MOVLW 50
        CPFSLT TEMP            ; skip if mark < 50
        INCF COUNT50, F        ; increment count

        DECFSZ N, F
        GOTO MARK_LOOP

; Compute average = SUM / 5
        MOVF SUM, W
        MOVWF TEMP
        MOVLW 0x05
        MOVWF N
        CLRF AVG

DIV_LOOP:
        SUBWF TEMP, F
        BTFSS STATUS, C
        GOTO DIV_END
        INCF AVG, F
        GOTO DIV_LOOP
DIV_END:

; Check if average >= 75
        MOVLW 75
        CPFSLT AVG
        GOTO LED_OFF
        GOTO LED_ON

LED_ON:
        MOVLW 0xFF
        MOVWF PORTB
        GOTO DONE

LED_OFF:
        CLRF PORTB

DONE:
        END
```