Key to Practical 2
Branches and Loops

Step 1

```
            org     $4

Vector_001  dc.l    Main

            org     $500

Main        clr.l   d1                  ; 0 -> D1
            move.l  #$80000007,d0       ; $80000007 -> D0.L (D0.W = $0007 = 7)
loop1       addq.l  #1,d1               ; D1 + 1 -> D1
            subq.w  #1,d0               ; D0.W - 1 -> D0.W ; Only D0.W is decremented
            bne     loop1               ; Branch if Z = 0 (D0.W ≠ 0)
                                        ; D1 = 7

            clr.l   d2                  ; 0 -> D2
            move.l  #$fe2310,d0         ; $fe2310 -> D0.L (D0.B = $10 = 16)
loop2       addq.l  #1,d2               ; D2 + 1 -> D2
            subq.b  #2,d0               ; D0.B - 2 -> D0.B ; Only D0.B is decremented
            bne     loop2               ; Branch if Z = 0 (D0.B ≠ 0)
                                        ; D2 = 8

            clr.l   d3                  ; 0 -> D3
            moveq.l #125,d0             ; 125 -> D0
loop3       addq.l  #1,d3               ; D3 + 1 -> D3
            dbra    d0,loop3            ; DBRA = DBF
                                        ; D0.W - 1 -> D0.W
                                        ; Branch if D0.W ≠ -1 (D0.W ≠ $FFFF)
                                        ; D3 = 126

            clr.l   d4                  ; 0 -> D4
            moveq.l #10,d0              ; 10 -> D0
loop4       addq.l  #1,d4               ; D4 + 1 -> D4
            addq.l  #1,d0               ; D0 + 1 -> D0
            cmpi.l  #30,d0              ; Compare D0 to 30
            bne     loop4               ; Branch if Z = 0 (D0.L ≠ 30)
                                        ; D4 = 20

            illegal
```

Step 2

```
VALUE       equ     18

            org     $4

Vector_001  dc.l    Main

            org     $500

Main        move.b  #VALUE,d1

            tst.b   d1          ; Set N and Z according to D1.B
            bne     next1       ; If Z = 0 (D1.B ≠ 0), then branch to Next1
            move.l  #200,d0     ; If not (D1.B = 0), 200 -> D0.L
            bra     quit        ; Exit
next1       bmi     next3       ; If N = 1 (D1.B < 0), then branch to Next3
            cmp.b   #$61,d1     ; If not (D1.B ≥ 0), D1.B is compared to $61 ($61 = 97)
            blt     next2       ; If D1.B < $61, then branch to Next2
            move.l  #400,d0     ; If not (D1.B ≥ $61), 400 -> D0.L
            bra     quit        ; Exit
next2       move.l  #600,d0     ; D1.B < $61, 600 -> D0.L
            bra     quit        ; Exit
next3       move.l  #800,d0     ; D1.B < 0, 800 -> D0.L

quit        illegal
```

1. What value is returned by the program when the `VALUE` label is set to 18?

   The program returns the value **600**.

2. What value is returned by the program when the `VALUE` label is set to –5?

   The program returns the value **800**.

3. What value is returned by the program when the `VALUE` label is set to 0?

   The program returns the value **200**.

4. What value is returned by the program when the `VALUE` label is set to 96?

   The program returns the value **600**.

Step 3

```
            org     $4

Vector_001  dc.l    Main

            org     $500

Main        ; Initialize D0.
            move.l  #-1,d0

Abs         ; Set Z and N according to D0.
            ; If D0 ≥ 0, then 0 -> N.
            ; If D0 < 0, then 1 -> N.
            tst.l   d0

            ; Branch to quit if N = 0 (if D0 ≥ 0).
            bpl     quit

            ; Otherwise N = 1 (D0 < 0).
            ; 0 – D0 -> D0
            neg.l   d0

quit        ; Stop the program.
            illegal
```

Step 4

```
            org     $4

Vector_001  dc.l    Main

            org     $500

Main        ; A0 points to the string.
            movea.l #STRING,a0

StrLen      ; Initialize the character counter to 0.
            ; (D0 = character counter).
            clr.l   d0

loop        ; Test if a character is null.
            ; A0 is incremented by one
            ; (it now points to the next character).
            tst.b   (a0)+

            ; If the tested character is null, it is the end of string.
            ; We can exit.
            beq     quit

            ; Otherwise, the counter is incremented by one.
            ; Then, branch to loop.
            addq.l  #1,d0
            bra     loop

quit        ; Stop the program.
            illegal

            org     $550

STRING      dc.b    "This string is made up of 40 characters.",0
```

Step 5

```
            org     $4

Vector_001  dc.l    Main

            org     $500

Main        ; A0 points to the string.
            movea.l #STRING,a0

SpaceCount  ; Initialize the space counter to 0.
            ; (D0 = space counter).
            clr.l   d0

loop        ; A character is loaded into D1.
            ; The MOVE instruction updates the flags
            ; in the same way as the TST instruction.
            ; Therefore :
            ; - If D1 ≠ 0, then 0 -> Z.
            ; - If D1 = 0, then 1 -> Z.
            ; The BEQ instruction can then be used.
            ; It jumps to quit if Z = 1 (if D1 = 0).
            move.b  (a0)+,d1
            beq     quit

            ; If the character in D1 is not a space,
            ; branch to loop.
            cmp.b   #' ',d1
            bne     loop

            ; Otherwise, the character is a space.
            ; The space counter is incremented.
            ; Then branch to loop.
            addq.l  #1,d0
            bra     loop

quit        ; Stop the program.
            illegal

            org     $550

STRING      dc.b    "This string contains 4 spaces.",0
```

Step 6

```
            org     $4

Vector_001  dc.l    Main

            org     $500

Main        ; A0 points to the string.
            movea.l #STRING,a0

LowerCount  ; Initialize the small-letter counter to 0.
            ; (D0 = small-letter counter).
            clr.l   d0

loop        ; A character is loaded into D1.
            ; The MOVE instruction updates the flags
            ; in the same way as the TST instruction.
            ; Therefore :
            ; - If D1 ≠ 0, then 0 -> Z.
            ; - If D1 = 0, then 1 -> Z.
            ; The BEQ instruction can then be used.
            ; It jumps to quit if Z = 1 (if D1 = 0).
            move.b  (a0)+,d1
            beq     quit

            ; If the ASCII code of the character is lower
            ; than that of 'a', the  character is not a small letter.
            ; So, branch to loop.
            cmp.b   #'a',d1
            blo     loop

            ; If the ASCII code of the character is higher
            ; than that of 'z', the  character is not a small letter.
            ; So, branch to loop.
            cmp.b   #'z',d1
            bhi     loop

            ; Otherwise, the character is a small letter.
            ; The small-letter counter is incremented.
            ; Then, branch to loop.
            addq.l  #1,d0
            bra     loop

quit        ; Stop the program.
            illegal

            org     $550

STRING      dc.b    "This string contains 29 small letters.",0
```