Key to Practical 5
Calculator (Part 2)

Step 1

```
Convert         ; If the string is empty,
                ; return false (error).
                tst.b   (a0)
                beq     \false

                ; (At this stage, the string is not empty.)
                ; If a character error occurs,
                ; return false (error).
                jsr     IsCharError
                beq     \false

                ; (At this stage, the string is not empty
                ;  and contains only digits.)
                ; If the integer value of the string is higher than 32,767,
                ; return false (error).
                jsr     IsMaxError
                beq     \false

                ; The string is valid. We can convert it
                ; and return true (no error).
                jsr     Atoui

\true           ; Return Z = 1 (no error).
                ori.b   #%00000100,ccr
                rts

\false          ; Return Z = 0 (error).
                andi.b  #%11111011,ccr
                rts
```

Step 2

```
Print           ; Save registers on the stack.
                movem.l d0/d1/a0,-(a7)

\loop           ; Load a character of the string into D0.
                ; If the character is null, it is the end of the string.
                ; We can exit the subroutine.
                move.b  (a0)+,d0
                beq     \quit

                ; Display the character.
                jsr     PrintChar

                ; Increment the column where the next character will be displayed
                ; and branch to \loop.
                addq.b  #1,d1
                bra     \loop

\quit           ; Restore registers from the stack and return from subroutine.
                movem.l (a7)+,d0/d1/a0
                rts
```

Step 3

```
NextOp          ; If the character is null (end of string),
                ; the string does not contain any operators.
                ; A0 points to the null character. Branch to \quit.
                tst.b   (a0)
                beq     \quit

                ; Compare successively the character to the 4 operators.
                ; If the character is an operator, branch to \quit.
                ; (A0 holds the address of the operator.)
                cmpi.b  #'+',(a0)
                beq     \quit

                cmpi.b  #'-',(a0)
                beq     \quit

                cmpi.b  #'*',(a0)
                beq     \quit

                cmpi.b  #'/',(a0)
                beq     \quit

                ; Go on with the next character.
                addq.l  #1,a0
                bra     NextOp

\quit           ; Return from subroutine.
                rts
```