PARAMETER PASSING AND CALLING SUBROUTINES

In this note we show how a subroutine call whose parameters are passed by reference, is implemented in Assembler.
Since the parameters are passed by reference, the changes of their values occur in the memory cells of the main program.

Consider the following subroutine definition:

```
      SUM(P1,P2,P3)
      P3 := P1 + P2
      END
```

and the following subroutine call in the main program:

```
         ...

      CALL SUM(A,B,S)
         ...
```

The subroutine call is translated into the following Assembler instructions:

```
         ...

      LOAX  7, *+2  [the address of this instruction + 2
                     is written into the index register 7]
      J     SUM
P1    RES   1
P2    RES   1
P3    RES   1
a        ...
```

(The choice of the index register 7 is arbitrary.)
The subroutine definition is translated into the following Assembler instructions:

```
SUM     LOA   1,     0 (7)  [the content of cell whose address is
                             (0 + content of the index register 7)
                             is written into the index register 1]
        ADD   1,     1 (7)
        STO   1,     2 (7)
        J     3 (7)         [jump to the instruction with label a]
        END   SUM
```

(Usually, the index register 1 is assumed to be the accumulator.)
==