How to compare any two numbers which are stored in R1 and R2 to see whether they are

1. R1 = R2
2. R1 > R2
3. R1 < R2
4. R1 <= R2

Make a comparison of R1 and R2.

執行

```
19   SUB R3 R1 R2      // R3=R1-R2
20   JUMPGT R3 3       //if jump to23, this means R1 > R2；if not jump, this means R3 <=0, R1 <= R2
21   JUMPNE R3 4       //if jump to 25, this means R1 < R2；if not jump, this mean R3 =0, R1 = R2
22   ….                // if can do this instruction, this means R1 = R2
23   ADD R5 R0 R0
24   ADDI R5 1
25   ROTATE R5 1 0
26   HALT
     END
```

The code below does the following task:

Read two integers from keyboard and put them into R1 and R2.

If R1 <= R2 then output R1+R2, otherwise output R1-R2

```
LOAD R1 254
LOAD R2 254
SUB R5 R1 R2
JUMPGT R5 2       // check whether R1 > R2
ADD R6 R1 R2      // It is R1 <= R2. So R6=R1+R2
STORE 255 R6
SUB R6 R1 R2      // It is R1>R2. So R6=R1-R2
STORE 255 R6
HALT
END
```

The code below does the following task:

Read an array of integers from keyboard. The array has K elements. K is a positive integer which must be read before all the other numbers are read from the keyboard. Then, the program should calculate the number of distinct integers stored in the array and display it on the monitor.

```
LOAD R1 254          // Read K from keyboard.
```

```
ADD R2 R0 R0        // R2 = 0
ADDI R2 10          // R2 = 10. The starting address of the array. It can be other number.
LOAD R5 254         // Read an element to be stored in the array from the keyboard
STOREI R2 R5        // Store the read element into the array with the address in R2
ADDI R2 1           // Update the address for storing next array element
ADDI R4 -1          // R4 = R4 -1. The number of elements yet to be read from keyboard reduced by 1
JUMPNE R4 -3        // If R4 != 0, jump to PC-3. That is jump to LOAD R5 254 to continue reading elements
ADD R3 R0 R0        // R3 = 0. Store the number of distinct elements in the array
ADD R4 R1 R0        // R4 = R1. Use R4 as in index.
ADD R2 R0 R0        // R2 = 0
ADDI R2 10          // R2 = 10
ADD R5 R0 R0        // R5 = 0
ADDI R5 11          // R5 = 11
READI R7 R2         // read an element from the array
READI R8 R5         // read another element from the array
SUB R9 R2 R5        // R9 = R2 – R9
JUMPNE R9 7
JUMPNE R3 3       // check whether R1 == R2 ?
STORE 255 R3    // It is R1 == R2
HALT
MOVE R4 R0    // At this point, R1 >R2 or R1<R2
ADDI R4 1
ROTATE R4 1 0
AND R4 R4 R3
JUMPNE R4 3 // check whether R1 > R2
STORE 255 R3    // It is R1 >R2
HALT
ADD R6 R1 R2    // It is R1<R2
STORE 255 R6
HALT
END
```