Prepare programs for:

(grade 3):
1. Evaluating a product of two natural numbers
2. Evaluating a quotient and reminder of two natural numbers.

(grade 4):
3. Evaluating the length of an array ended with a specified byte (e.g. '0') [c]
4. Finding the greatest value in an array of a given length. [a]
5. Finding the lowest value in an array ended with a specified byte (e.g. '0'). [c]
6. Searching an array for the greatest element and storing its value and index in separate variables (allocated and labeled memory cells). [a, b, c]
7. Searching an array for the lowest element and storing its value and index in separate variables (allocated and labeled memory cells). [a, b, c]
8. Searching for a value specified in a separate variable and storing the index of its first occurrence. [a, b, c]
9. Searching for a value specified in a separate variable and counting a number of occurrences. [a, b, c]

(grade 5):
10. Incrementing all elements of an array. [a, b, c]
11. Decrementing all elements of an array. [a, b, c]
12. Swapping pairs of an array's elements. The length of the array is a given even value. [a]
    Ex.: ABCDEF -> BADCFE
13. Swapping of an array's elements. The length of the array is a given value. [a]
    Ex.: ABCDE -> EDCBA

(grade 6):
14. Evaluating NWD (the greatest common divider) of two natural numbers using the Euclides algorithm
15. Evaluating NWW (the smallest common multiple) of two natural numbers
16. Evaluating factorial value for a given natural number.
17. Exponentiation of two natural numbers.
18. Evaluating a determinant of 3 x 3 matrix. The matrix is stored as a one-dimensional table.
19. Evaluating a value of symbol $\binom{n}{k}$.

- **[a] a number of elements specified in a separate variable,**

```
SIZE:   RST 3       ; number of elements: 3
ARRAY:  RST 1
        RST 2
        RST 3
```

- **[b] a number of elements given as its first element,**

```
ARRAY:  RST 3       ; number of elements: 3
        RST 1
        RST 2
        RST 3
```

- **[c] use of a special characteristic value to indicate the end of an array (especially useful in case of the "find a pattern" type of task).**

```
CharValue: RST 0
ARRAY:  RST 1
        RST 2
        RST 3
        RST 0       ; end of an array (string)
```

Additional information:

List of instructions with mnemonic and code

Operation	Mnemonic	Machine instruction
(AK) + ((AD)) -> AK	DOD	001
(AK) - ((AD)) -> AK	ODE	010
((AD)) -> AK	POB	011
(AK) -> (AD)	LAD	100
(AD) -> L	SOB	101
(AD) -> L , if Z = 1	SOM	110
(L) + 1 -> L , if Z = 0		
STOP	STP	000

example – Evaluating a sum of values of all elements of an array:

```
loop:   POB size
        ODE one
        SOM end          // if AK [ACC] is negative then jump to label   'end'
        LAD size

        POB sum
ins:    DOD array
        LAD sum
            POB ins
            DOD one
            LAD ins
        SOB loop

end:    POB sum
        STP

array:  RST 1
        RST 3
        RST 1
        RST 5
size:   RST 4
sum:    RST 0
one:    RST 1
```