# Programming 2 - Laboratories: Task 12

In this task you have to implement class templates and create functions for revers polish notation (RPN) conversion and calculation.

## Part 1 (3 points)

The `linkedQueue` class represents the class of a queue with a double-ended access. Convert the `linkedQueue` class to a class template for general type `T`.

## Part 2 (2 point)

Abstract template class `stackADT` defines public interface for classes realizing stack.

Implement `linkedStack` class template for general type `T`, which realise the idea of the stack based on the queue. `linkedStack` class template should be privately inherited from the `linkedQueue` template class, as well as publicly inherited from the `stackADT` template class:

```
template <typename T = int>
class linkedStack : public stackADT<T>, private linkedQueue<T>
{
public:
        ….
};
```

In implementation provide only necessary elements.

## Part 3 (1,5 points)

Implement the template function

```
template <typename T>
T rpn_value(string rpn_expression)
```

calculating the value of the expression given in reverse Polish notation (RPN). RPN is a way to write arithmetic expressions without the use of parentheses, in which the operation symbol follows arguments. Type `T` should be an arithmetic type.

The algorithm for calculating the value of an expression:

1. Analyse the expression element by element.
2. If analysed element is:
    a. constant - add to the stack;
    b. operator - remove from the stack the right number of arguments for the given operator, perform calculations on them, add the result to the stack;
3. If analysed expression:
    a. has not been exhausted - go back to the first point;
    b. has been exhausted - the value on the stack is the result of the calculation.

Example:
Tested expression:
6 3 / 2 5 + *

The individual steps of the algorithm are presented in the table below:

| Krok | Wejście | Operacja | Stos |
|---|---|---|---|
| 1 | 6 | | 6 |
| 2 | 3 | | 6 3 |
| 3 | / | 6 / 3 | 2 |
| 4 | 2 | | 2 2 |
| 5 | 5 | | 2 2 5 |
| 6 | + | 2 + 5 | 2 7 |
| 7 | * | 2 * 7 | 14 |

## Part 4 (1,5 points)

Implement the function:

```
string rpn(string symbolic_expression)
```

converting symbolic expression to reverse Polish notation (RPN). The algorithm of exchanging the expression:

The algorithm for conversion is as follow:

1. Analyse the expression element by element – convert each element to `expression` object (make use of `expression` class, which is given).
2. If analysed expression is:
    a. constant - pass it to the output;
    b. operator:
        i. if the priority of the tested operator is higher than the priority of the operator occupying the top of the stack or if the stack is empty - add it (expression) to the stack;
        ii. if there is an operator with higher or equal priority on top of the stack - read from the stack and send to the output all operators with a higher or equal priority until operator with lower priority is not met on the top of the stack.
    c. parenthesis
        i. if you found the opening parenthesis - add it to the stack;
        ii. if you reach the closing parenthesis: remove all operators from the stack and pass them to the output until you reach the opening parenthesis; do not pass parenthesis to the output.
3. If analysed expression:
    a. has not been exhausted - go back to the first point;
    b. has been exhausted - read all operators from the stack and pass them to the output;

To properly execute the task, it is necessary to set priorities of the operators (see class `expression`).

*Example program output:*

```
 ---------- STAGE_1 (3 Pts) ----------
1 2 3 4 5
Front value: 1
Rear value: 5
Cannot print an empty queue

(1,1) (2,1) (3,1) (4,1) (5,1)
Front value: (1,1)
Rear value: (5,1)
Cannot print an empty queue

 ---------- STAGE_2 (2 Pts) ----------
4 3 2 1
Cannot remove from an empty queue

(4,1) (3,1) (2,1) (1,1)
Cannot remove from an empty queue

 ---------- STAGE_3 (1,5 Pts) ----------
3 + 5 * 2  in RPN is:  3 5 2 * + = 13
(3 + 5) * 2  in RPN is:  3 5 + 2 * = 16
6 / 3 * (2 + 5)  in RPN is:  6 3 / 2 5 + * = 14
3 / 6 * (0.25 + 0.25)  in RPN is:  3 6 / 0.25 0.25 + * = 0.25

 ---------- STAGE_4 (1,5 Pts) ----------
3 5 2 * +  = 13
3 5 + 2 *  = 16
6 3 / 2 5 + *  = 14
3 6 / 0.25 0.25 + *  = 0.25
```