EADS Task 2 (10 points)

Design and implement a class template:

```
template <typename Data>
class Ring {
// ...
// implemented using a doubly-linked ring
// and Iterator
}
```

The class should be widely commented in the source file (instead of a report) to explain the design and the implementation. The class should be **universal** and **complete**. Extensive testing is expected and is a subject of evaluation. The methods in class Ring should **use only objects of Iterator** for denoting a place in the structure.

To test flexibility of the class template above implement the following function template:

```
template <typename Data>

std::pair<Ring<Data>, Ring<Data>>

split(const Ring<Data> &source,

int startIndex, int length, bool direction,

int step1, bool direction1, int step2, bool direction2);
```

The function template split should produce two rings from a single ring (**source**) by repeating taking alternatively **step1** elements for **ring1** and **step2** elements for **ring** 2 until **length** elements is used in total. alternative subsequences of lengths length1 and length2 respectively. The Boolean parameters denotes directions (true – clockwise, false – counterclockwise) in which appropriate sets of elements are used. **std::pair** can be found in **<utility>**

The solution has to be submitted to appropriate lab supervisor by e-mail no later than on Monday, December 7th, 2020 before 9:00 am. Each subsequent week of delay is punishable by subtraction of 2 points.

The solutions of Task 2 will be discussed individually with each student on December 9th, 2020 via calls on MS Teams.

Let's have an example with rings
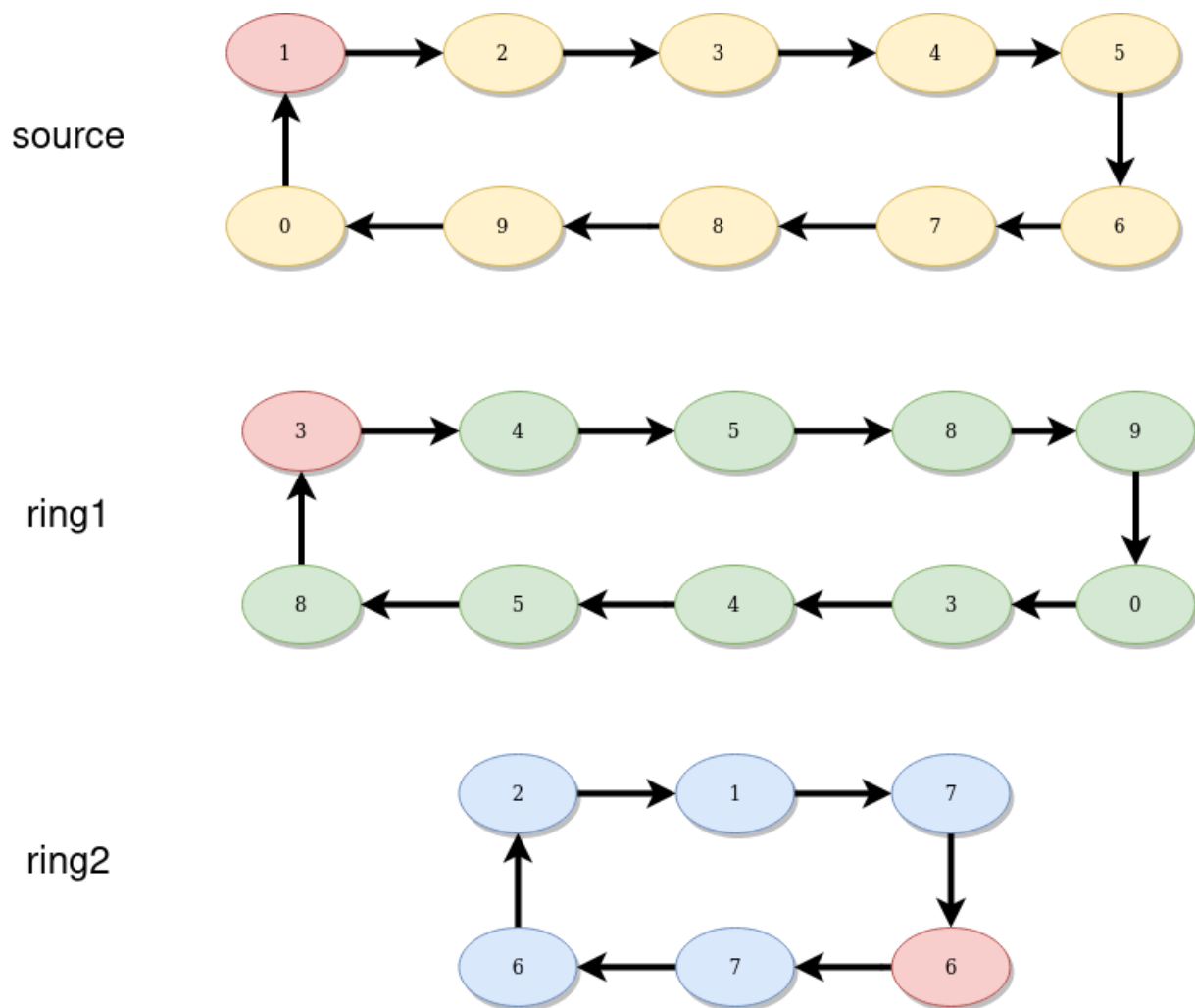
The drawing above denotes ring1 and ring2 produced by function split from ring source by applying the following parameters:

```
split (
source, 2, 16, true, // start +2 elements, use 16 in total,
                     // clockwise
3, true,       // use 3 elements clockwise
2, false);      // use 2 elements counterclockwise
```

Element pointed by **any** is denoted by a red frame.