Algorithms and Data Structures

laboratory

Task #1 LAB-104 (term 2025Z)

Task #1 deadline for class LAB-104 is November 5, 2025.

To get points from the Laboratory, student should:

- send back TEAMS assignment with source files attached (only three source files),
- the assignment should be handed in (sent back) before deadline,
- the source files names (attached to the assignment) should be:
  sequence.hpp, split.hpp and main.cpp,
- present (live in lab) the project to the supervisor at the meeting on November 5, 2025.

1 PART1 - DESIGN A CLASS

Design a class to represent a collection (implemented as a singly linked list). Write unit tests for the designed class, at least one test per method.

```
template <typename Key, typename Info>
class Sequence
{
	// implemented as singly linked list
	// the Sequence is not ordered neither by Key nor Info

public:
	Sequence();

	Sequence(const Sequence& src)
	{
		// initialize properly data members
		// probably as in default constructor
		*this = src;
	}

	~Sequence();

	Sequence& operator=(const Sequence& src);

	unsigned int size() const;
	void push_front(const Key& key, const Info& info); // keys are not unique!
	bool pop_front();

	// and the rest of the interface, according to your own design

private:

	// private data members and private member functions

};
```

2 PART 2 – WRITE TWO ADDITIONAL FUNCTION TEMPLATES

Write two additional, external function templates (not methods in the Sequence class) and provide unit tests for them.

```
//Function #1 - creates two sequences by splitting the existing one
//(moves elements from the source collection seq
// to the target collections: seq1, seq2
// the starting element in the source sequence is specified by a position number index
template <typename Key, typename Info>
void split_pos(const Sequence <Key, Info>& seq, int start_pos,
               int len1, int len2, int count,
               Sequence <Key, Info>& seq1, Sequence <Key, Info>& seq2)
{      …
}
// Function #2 - creates two sequences by splitting the existing one
//(moves elements from the source collection seq
// to the target collections: seq1, seq2
// the starting element in the source sequence is specified by a Key value
// and occurrence number
template <typename Key, typename Info>
void split_key(const Sequence <Key, Info>& seq, const Key& start_key, int start_occ,
               int len1, int len2, int count,
               Sequence <Key, Info>& seq1, Sequence <Key, Info>& seq2)
{      …
}
```

Example 1 for split_pos:

Parameters:

```
seq={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24},
start_pos=2, len1=2, len2=3, count=4
```

Result:

seq={0,1,22,23,24}

seq1={2,3,7,8,12,13,17,18}

seq2={4,5,6,9,10,11,14,15,16,19,20,21}

Example 2 for split_pos:

Parameters:

```
seq={0,1,2,3,4,5,6,7,8,9,10}, start_pos=2, len1=2, len2=3, count=4
```

Result:

seq={0,1}

seq1={2,3,7,8}

seq2={4,5,6,9,10}

Example for split_key:

Parameters:

```
seq={0,1,2,3,4,5,6,4,8,9,4,11,12,2,14,15,11,17,23,19,20,21,22,23,24},
start_key=4, start_occ=2, len1=3, len2=2, count=2
```

Result:

```
seq={0,1,2,3,4,5,6,17,23,19,20,21,22,23,24}
seq1={4,8,9,12,2,14}
seq2={4,11,15,11}
```