EADS Task 1 (10 points)

Design and implement a class template:

```
template <typename Key, typename Info>
class Sequence {
// ...
// implemented using a singly-linked list
}
```

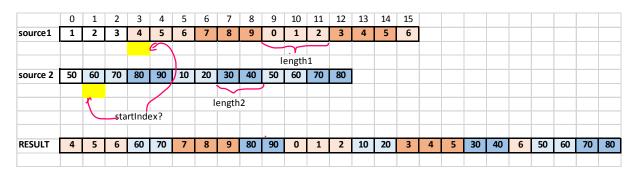
To test flexibility of the class template above implement the following function template:

```
template <typename Key, typename Info>
Sequence<Key, Info> shuffle (
const Sequence<Key, Info> & source1, int startIndex1, int length1,
const Sequence<Key, Info> & source2, int startIndex2, int length2,
int limit)
```

The function template shuffle should produce a sequence from two input sequences by interleaving subsequences of lengths length1 and length2 respectively. The subsequences should start at positions startIndex1 and startIndex2 respectively and continue till the ends of source sequences. You shouldn't change the signature of the Sequence class nor of the shuffle method.

Let's have an example with sequences, where keys are presented with colours distinguishing the source sequences

```
RESULT = shuffle (
source1, 3, 3,
source2, 1, 2,
30)
```



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 class template may be equipped with an iterator, but this is not an obligatory part (iterator will be required with Task 2).

The solution has to be submitted via teams assignment no later than on Monday, November 23rd, 2020 before 9:00 am. Each subsequent week of delay is punishable by subtraction of 2 points.

The solutions of Task 1 will be discussed individually with each student on November 25th, 2020 via calls on MS Teams.

Checklist before you submit:

- Is my code compiling?
- Are there any compilation warnings? There shouldn't be. Moreover, you should compile with these flags and remove the resulting warnings (if any): -Wall -Wextra -pedantic -Werror
- Is my code working correctly?
- Have I tested my code in a reasonable fashion? Reasonable here means that you either use some kind of unit testing framework or write your own solution. Please note that cout-based testing is **NOT** satisfactory you should only cout some information if tests do not pass.
- Is my code reusable and am I reusing it in shuffle implementation? Make sure that you use methods/operators/constructors that you have already implemented to mitigate code repetition.
- Is my code clean and neatly formatted?
 - o remove any unnecessary commented-out code (aka dead code),
 - make sure that your naming is acceptable (that means sensible variable/method names)
- Am I using references correctly? Make sure that you pass-by-reference wherever you can rather than by copy/move (except places in which it makes sense to do so)
- Am I using const keyword correctly? Make sure that you mark references/methods/objects
 as const wherever it makes sense to do so (most of your references and methods should
 probably be const)
- Is my code memory-leak free? You can use valgrind to check this property.
- Is my code undefined behavior free? Your can compile your code with the fsanitize=undefined flag and run your tests if it doesn't crash then you're most likely good to go.

Files to submit:

• Source code (.cc/.cpp/.hpp), including tests

Files **NOT** to submit:

- Editor configuration files/directories
- binaries/object files

Please make sure that you click *submit* (upper right corner in teams app) after you have selected all files – otherwise the submission won't be registered.