

# 02393 C++ Programming Exercises

Week 10, November 7, 2016

**Hand-in via `dtu.codejudge.net`, before November 14, 5pm**

**Doubly-Linked Lists** The goal of the exercise is to implement a class `List` of doubly-linked lists. The specification of the class is provided in file `doubly-linked.h`, available in CodeJudge. Such file is imported by the test programs. Your goal is to implement the class in a file `doubly-linked.cpp`.

The class `List` implements lists using the following structures for list elements:

```
struct Node {
    int val;
    Node * next;
    Node * prev;
};
```

Each `Node` contains a value (an integer) and two pointers: one meant to point to the next element of the list and one meant to point to the previous element in the list.

The class provides the following methods that you need to implement:

- A constructor and a destructor, with no arguments;
- `void insert(int n)`: insert an integer **at the end of the list**;
- `void reverse()`: reverse the list;
- `void print()`: print the list to `cout` in the same format as the input (i.e. integers separated by spaces);

**Extra Challenge** Implement the class so that both `insert` and `reverse` work in constant time.