

Labwork 1

Q1 (50pts): Write a **non-member** C++ function which takes an `IntSLList` reference as a parameter, and returns the sum of all elements in the list. You will get only partial grade if the contents of the list is changed after the function call. Prototype of your function should exactly be:

```
int list_sum(IntSLList & list);
```

Important: Put your work in a file named `list_sum.cpp`. Your work should compile & run along with the example main file provided to you. You can compile multiple cpp files using:

```
g++ main.cpp list_sum.cpp
```

Send only `list_sum.cpp`, and do not send a main function. Also, make no changes to the `IntSLList` class for this question. You can use existing functions from the `IntSLList` class.

Q2 (50pts): You will see that there is a member function definition in `IntSLList.h` named `remove_odds`, but there is no code associated with it in the cpp file. Your task is to implement this member function.

The function removes all odd-indexed elements from the linked list. For example, if the contents of the linked list is

7 17 27 37 47 57

before the function call, it should become

17 37 57

after the call. You are not allowed to call any pre-existing functions from the `IntSLList` class, but you can check the code and get inspiration. Send only your modified `IntSLList.cpp` file for this question.

Also, do not forget about corner-cases and memory leaks 😊