Oddly Specific Deletion

> This is a **warmup** exercise. It is **not compulsory**, and may be completed **individually or with your lab partner**.

Time for some more linked list fun. This time, let's get rid of all the nodes with odd-numbered values.

As you no doubt recall, a linked list is a collection of *nodes*, where each node stores a value of a particular type, as well as the location of the following node.

We also have a `list` structure, which lets us keep track of the first element in the list.

You'll need different types of `node`s for various different types of data. In your assignment, you probably have a few different types of linked lists.

As you saw last week, a linked list of integers would have the following definition:

```
typedef struct _node *Node;
typedef struct _list {
    Node head;
} list;
typedef struct _node {
    int value;
    Node next;
} node;
```

We'll use the same `list.h` as we had last week, which defines these types.

Download `list.h`, or copy it into your current directory on a CSE system by running

```
$ cp /web/cs1511/17s2/week12/files/list.h .
```

For this exercise, create a file *deleteOdd.c*; in it, we're going to implement a function, `deleteOdd`, which deletes odd-valued nodes from the list.

Here's the prototype:

```
void deleteOdd (List l);
```

So, if I have this list:

$$l := 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow \varnothing$$

... and I call `deleteOdd` on l, I should have the following list:

$$l := 2 \rightarrow 4 \rightarrow \varnothing$$

You should write your own tests in a separate file; `deleteOdd.c` should *not* contain a `main`.

To run Styl-o-matic:

```
$ 1511 stylomatic deleteOdd.c
Looks good!
```

You'll get advice if you need to make changes to your code.

Submit your work with the *give* command, like so:

```
$ give cs1511 wk12_deleteOdd
```

Or, if you are working from home, upload the relevant file(s) to the `wk12_deleteOdd` activity on Give Online.