Computer Programming 2
Lab worksheet (week 13) Linked Lists

This week we will use the file print.c from the linked list lecture. This is on moodle in week 12 or next to this file on moodle in week 13.

The output of print.c is:

```
Length is: 4
{ 0, 1, 2, 3 }
```

Make a copy of print.c called print_solution.c

Question 1. Add a function to print_solution.c called value_at.

- This function should get the value at node `n` of the linked list (the first node is node 1, etc. – in other words let's not use 0 for the first node).
- This function should return an int (the value at position `n`) and have two parameters: `Node *head` and `int n`
    - So its definition is:

```
int value_at(Node *head, int n)
```

- If `n` is less than 1, or greater than the length of the list, the function should return -1, which for us indicates an error. Note, this is not the best way to do this, but for now it is OK.
- Test this function by adding code to the main function to print the value at different positions of the list
- The value at the specified node number `n` should be printed to the screen

Example output:

```
Length is: 4
{ 0, 1, 2, 3 }
Question 1: The value at node 2 is: 1
```

Question 2. Add a function to print_solution.c called add_element_at_end.

- This function should insert a new element at the end of the list (in a new node)
- This function should return void (as the head of the list does not change, unlike when we added a new node at the beginning of the list, and have two parameters: `Node *head` and `int element`, so its definition is:

```
void add_element_at_end(Node *head, int element)
```

- Test this function by adding code to the main function to add a new element to the end of the list and print out the new length of the list and the new list elements

Example output:

```
Length is: 4
{ 0, 1, 2, 3 }
Question 1: The value at node 2 is: 1
Question 2: Adding value 4 at end of list in a new node – node 5
Length is: 5
{ 0, 1, 2, 3, 4 }
```