Christopher McCabe

ML Lab

1.

```
 val it = [4, 2, 5, 2, 6, 10]: int list
```

2. The function ‘f’ takes an int list and increases each entry by 1.
3. The first line has this function return an empty list if the passed parameter is an empty list.
4. The second line, and alternative version of the function, breaks the passed list into its head and its tail. It adds one to its head, and then concatenates the result with the tail – but first, the tail is passed recursively back into the f function.
5.

```
 Question 5: false
 val it = (): unit
```

6. Tail recursion.
7.

```
 { best times, it was the worst of times }
 val it = (): unit
```

8. Tail recursion is easier to write for most computer science purposes and their left-hand derivation techniques.
9.

```
 { green eggs and ham }
 val it = (): unit
```

10.

```
 { macaroni and }
 val it = (): unit
```

11. I really enjoy the different type of thinking required to write for ML. It’s a wonderful exercise in recursive practices.
12. The least enjoyable part of ML is the environment, I’m not enjoying the difficulty at running each line of code.
13. I don’t really enjoy its due date but otherwise it’s fine.
14. One to two.
15. I find it unlikely considering its situational usage and it’s another program to download, but I definitely enjoy