Common Lisp Practicum Test

Questions

Question 1

NOTE:

- You are required to write the solutions for the parts of this question in the Lisp program file ~**/pt/q1.lisp**
- You may create helper functions in your program file.
- You must not use or refer to the following Lisp built-in function(s) and symbol(s): **COUNT**, **MEMBER**. The penalty for doing so is a deduction of 80% on the score of your solutions for this question.

Part 1: Write a function `count-occurrences` that takes an element and a list as arguments and returns the number of times the element appears in the list.

```
CL-USER> (COUNT-OCCURRENCES 3 '(1 2 3 3 3 4))
3
CL-USER> (COUNT-OCCURRENCES 'A '(A B A C A))
3
CL-USER> (COUNT-OCCURRENCES 5 '(1 2 3 4))
0
```

Part 2: Write a function `contains-all?` that takes two lists as arguments and returns `T` if all elements of the first list are contained in the second list, and `NIL` otherwise.

```
CL-USER> (CONTAINS-ALL? '(1 2) '(1 2 3 4))
T
CL-USER> (CONTAINS-ALL? '(1 5) '(1 2 3 4))
NIL
CL-USER> (CONTAINS-ALL? 'NIL '(1 2 3))
T
```

Question 2

NOTE:

- You are required to write the solutions for the parts of this question in the Lisp program file ~**/pt/q2.lisp**
- You may create helper functions in your program file.
- There are no restrictions in the use of Lisp built-in functions or symbols in the parts of this question.

Part 1: Write a function `reverse-list` that takes a list as an argument and returns a new list that is the reverse of the original list.

```
CL-USER> (REVERSE-LIST '(1 2 3 4))
(4 3 2 1)
CL-USER> (REVERSE-LIST '(A B C))
(C B A)
CL-USER> (REVERSE-LIST 'NIL)
()
```

Part 2: Write a function `palindrome?` that takes a list as an argument and returns `T` if the list is a palindrome (reads the same forwards and backwards), and `NIL` otherwise. You may not use `REVERSE` or `NREVERSE`.

```
CL-USER> (PALINDROME? '(1 2 3 2 1))
T
CL-USER> (PALINDROME? '(A B C D))
NIL
CL-USER> (PALINDROME? 'NIL)
T
```