Question 1

Write down what the following expressions evaluate to; try to work them out before trying on the computer.[1]

(a) `(cons 2)`

(b) `(cons 2 NIL)`

(c) `(cons 3 2)`

(d) `(cons NIL NIL)`

(e) `(cons (1 2) NIL)`

(f) `(cons '(1 2) NIL)`

(g) `(cons (A B) NIL)`

(h) `(cons ('A 'B) NIL)`

(i) `(cons '(A B) NIL)`

(j) `(cons '(A B) '(C D))`

(k) `(list 1 4)`

(l) `(list 1 '4)`

(m) `(list '1 4)`

(n) `(list 'A B)`

(o) `(list 'A 4)`

(p) `(list 'A 'B)`

(q) `('list 1 4)`

(r) `(+ 3 '4)`

(s) `('+ 1 4)`

(t) `(list 3 'times '(- 5 2) "is 9")`

(u) `(eval (list '+ 1 2))`

Question 2

Define a function named `insert-2nd`, which takes a list and an element, and gives back another list where the element is inserted after the first element of the given list. Here is a sample interaction:[2]

```
* (insert-2nd 'b '(a c))

(A B C)
* (insert-2nd '(b k) '(a c))

(A (B K) C)
*
```

[1] For errors just mark is as an error, no need to specify the error itself.

[2] In this and the next question, do not worry about what happens if the input list is shorter than 2. However try such inputs and make sure that you understand why your program behaves that way.

Question 3

Define a function named `replace-2nd`, which *replaces* the element at the 2nd position.

Question 4

Define a function `swap`, that takes a two element list and switches the order of the elements.

Question 5

The function `wrap-2` takes a two element list, and wraps each element inside a list:

```
* (wrap-2 '(a b))

((A) (B))
* (wrap-2 (wrap-2 '(a b)))

(((A)) ((B)))
* (wrap-2 (wrap-2 (wrap-2 '(a b))))

((((A))) (((B))))
*
```

Write two versions of `wrap-2`:

(a) one using `list`

(b) the other using `cons`

Either hand in your assignment in class or send it to Fırat before the class on Mar 14.