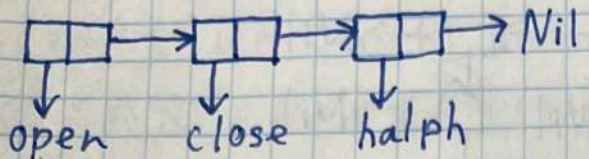
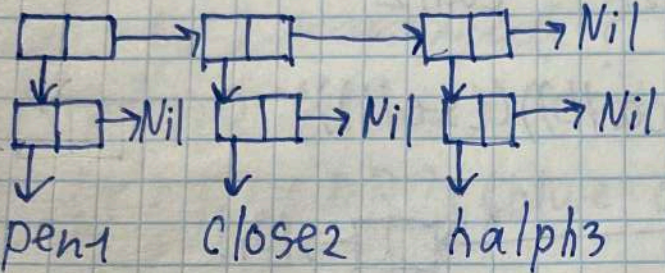
w1. Ковалёв К.
ИУ7-63Б

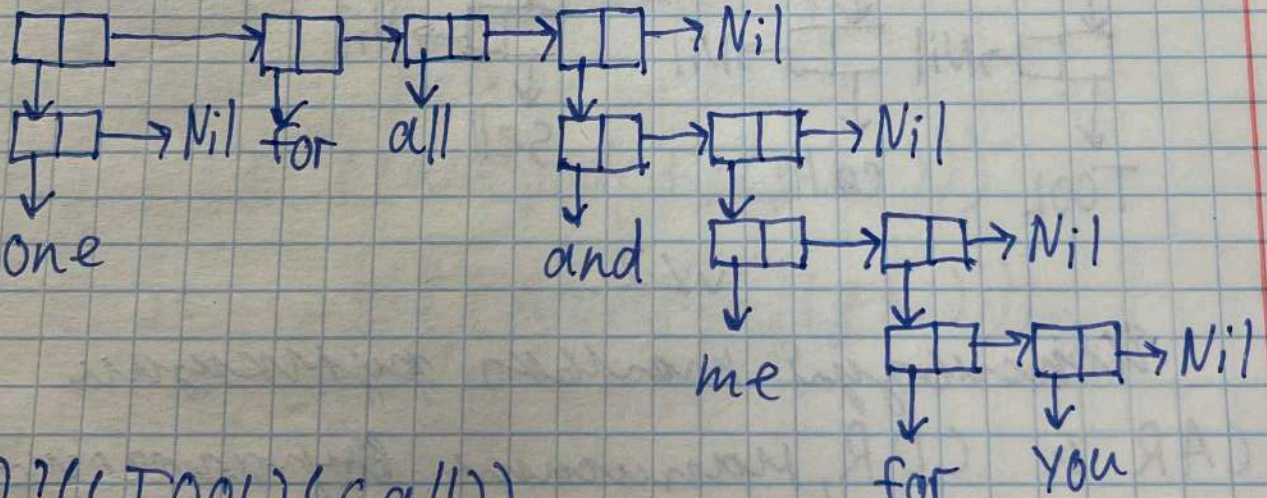
1) '(open close halph)



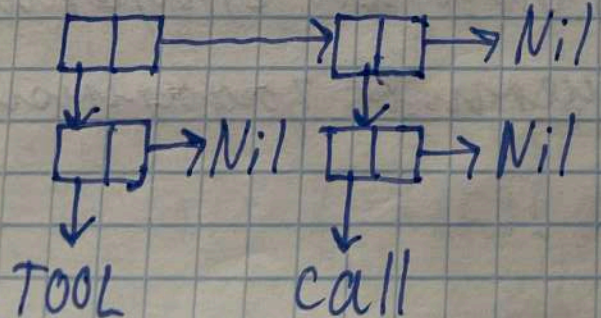
2) '((open1)(close2)(halph3))



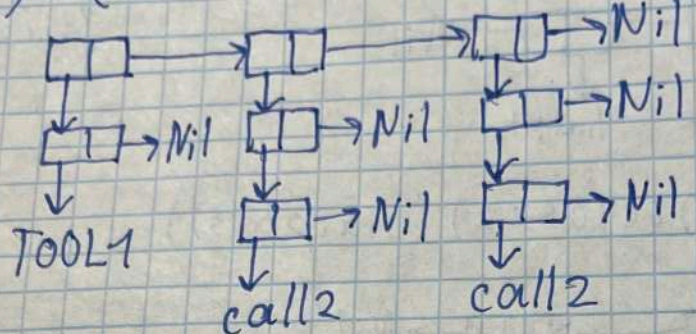
3) '((one) for all (and (me (for you))))



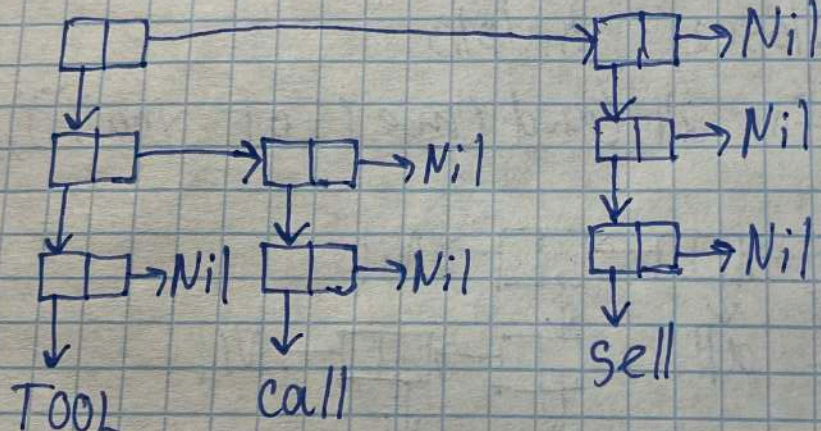
4) '((TOOL)(call))



5) '((TOOL 1)((call 2))((sell)))



6) '(((TOOL)(call))((sell)))



№2.

Используя только функции CAR и CDR, написать выражения, возвращающие 1) второй 2) третий 3) четвёртый элементы заданного списка.

1) (CAR (CDR '(a b c d)))
2) (CAR (CDR (CDR '(a b c d))))
3) (CAR (CDR (CDR (CDR '(a b c d)))))

№3.

Что будет в результате вычисления выражений?

a) (CAADR '((blue cube) (red pyramid)))
(CAR (CAR (CDR '((blue cube) (red pyramid)))))
(CAR (CAR '((red pyramid))))
(CAR '(red pyramid))
Ответ: red

b) (CDAR '((abc) (def) (ghi)))
(CDR (CAR '((abc) (def) (ghi))))
(CDR '(abc))
Ответ: Nil

c) (CADR '((abc) (def) (ghi)))
(CAR (CDR '((abc) (def) (ghi))))
(CAR '((def) (ghi)))
Ответ: (def)

d) (CADDR '((abc)(def)(ghi)))
(CAR(CDR(CDR'((abc)(def)(ghi)))))
(CAR(CDR'((def)(ghi))))
(CAR'((ghi)))
Ответ: (ghi)

№4.

Запишите результат вычисления выражений и объясните как он получен:

1) (list 'Fred 'and 'Wilma) =>
(Fred and Wilma)
2) (list 'Fred '(and Wilma)) =>
(Fred (and Wilma))
3) (cons Nil Nil) => (Nil)
4) (cons T Nil) => (T)
5) (cons Nil T) => (Nil.T)
6) (list Nil) => (Nil)
7) (cons '(T) Nil) => ((T))
8) (list '(one two) '(free temp)) =>
((one two)(free temp))

Ковалец К.
ИУ7-63Б

9) (cons 'Fred '(and Wilma)) =>
(Fred. and Wilma)) =>
(Fred and Wilma)

10) (cons 'Fred '(Wilma)) =>
(Fred Wilma)

11) (list Nil Nil) => (Nil Nil)

12) (list T Nil) => (T Nil)

13) (list Nil T) => (Nil T)

14) (cons T (list Nil)) =>
(cons T (Nil)) =>
(T Nil)

15) (list '(T) Nil) => ((T) Nil)

16) (cons '(one two) '(free teamp)) =>
((one two) free teamp)

№5.

Написать лямбда-выражение и соответствующую функцию:

1) написать функцию (f ar1 ar2 ar3 ar4), возвращающую список:
((ar1 ar2)(ar3 ar4))

```
(defun f(ar1 ar2 ar3 ar4)(list
        (list ar1 ar2)(list ar3 ar4)))

(defun f (ar1 ar2 ar3 ar4)(cons
        (cons ar1(cons ar2 nil))
        (cons(cons ar3 (cons ar4 nil))nil)))
```

2) написать функцию (f ar1 ar2), возвращающую ((ar1)(ar2))

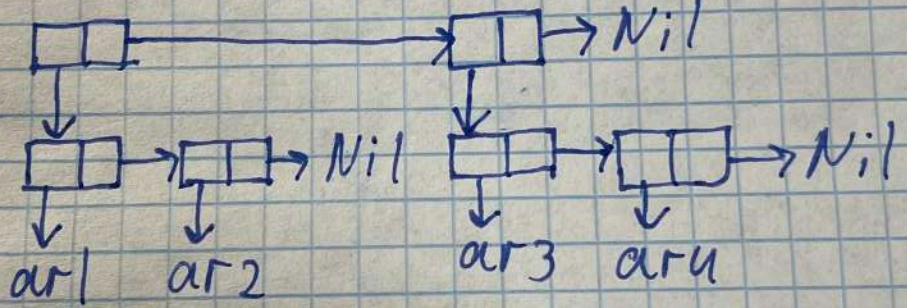
```
(defun f(ar1 ar2) (list
        (list ar1) (list ar2)))
(defun f(ar1 ar2) (cons
        (cons ar1 nil)
        (cons (cons ar2 nil)nil)))
```

3) написать функцию (f ar1), возвращающую (((ar1))).

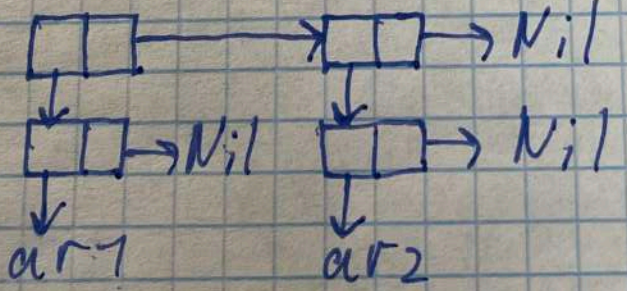
```
(defun f(ar1) (list (list (list ar1))))
(defun f (ar1)
   (cons(cons(cons ar1 nil) nil)nil))
```

4) представить результаты в виде списочных ячеек.

а) ((ar1 ar2) (ar3 ar4))



б) ((ar1)(ar2))



в) (((ar1)))