Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ» (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

О Т Ч Е Т

по лабораторной работе № 6

Дисциплина: Функциональное и логическое программирование

Студент	ИУ7-62Б		Е.В. Брянская
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			Н.Б.Толпинская
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2021

Лабораторная работа №6

Из лаб 4: 1-5

№1 Чем принципиально отличаются ф-ции cons list append?
Пусть (setf lst1 '(a b))
(setf lst2 '(c d))

Каковы результаты вычисления следующих выражений?

(cons lst1 lst2)
(list lst1 lst2)
(append lst1 lst2)

cons – принимает только 2 аргумента, является базовой функцией, создает только одну списковую ячейку и распределяет указатели на эл-ты

list – создает столько списковых ячеек, сколько аргументов, car-ук-ли этих ячеек устанавливаются на аргументы, с помощью cons ук-ли связывают списковые ячейки, последний ук-ль выставляется в Nil

append – принимает произвольное количество аргументов, на вход подаются существующие списки, создает копию всех списков, кроме последнего (копирует списковые ячейки)

(cons lst1 lst2) = ((a b) c d)
(list lst1 lst2) = ((a b) (c d))
(append lst1 lst2) = (a b c d)

№2 Каковы результаты вычисления след. выр-ний?
(reverse ()) = Nil
(last ()) = Nil
(reverse '(a)) = (a)

(2) (last '(a)) = (a)
(reverse '((a b c))) = (c b a)
(last '((a b c))) = (c)

(№3) Написать, по крайней мере, два варианта функции, которая возвращает последний элемент своего списка-аргумента.

① (defun last-elem (lst)
    (cond ((cdr lst) (last-elem (cdr lst)))
          (t (car lst))
)
)

② (defun last-elem (lst)
    (car (last lst))
)

③ (defun last-elem (lst)
    (car (reverse lst))
)

(№4) Написать, по крайней мере, два варианта функции, которая возвращает свой список-аргумент без последнего элемента.

① (defun without-last (lst)
    (cond ((cdr lst) (cons (car lst) (without-last (cdr lst))))
          (t (setf (car lst) Nil))
)
)

(2)
```
(defun without-last (lst)
    (reverse (nthcdr 1 (reverse lst)))
)
```

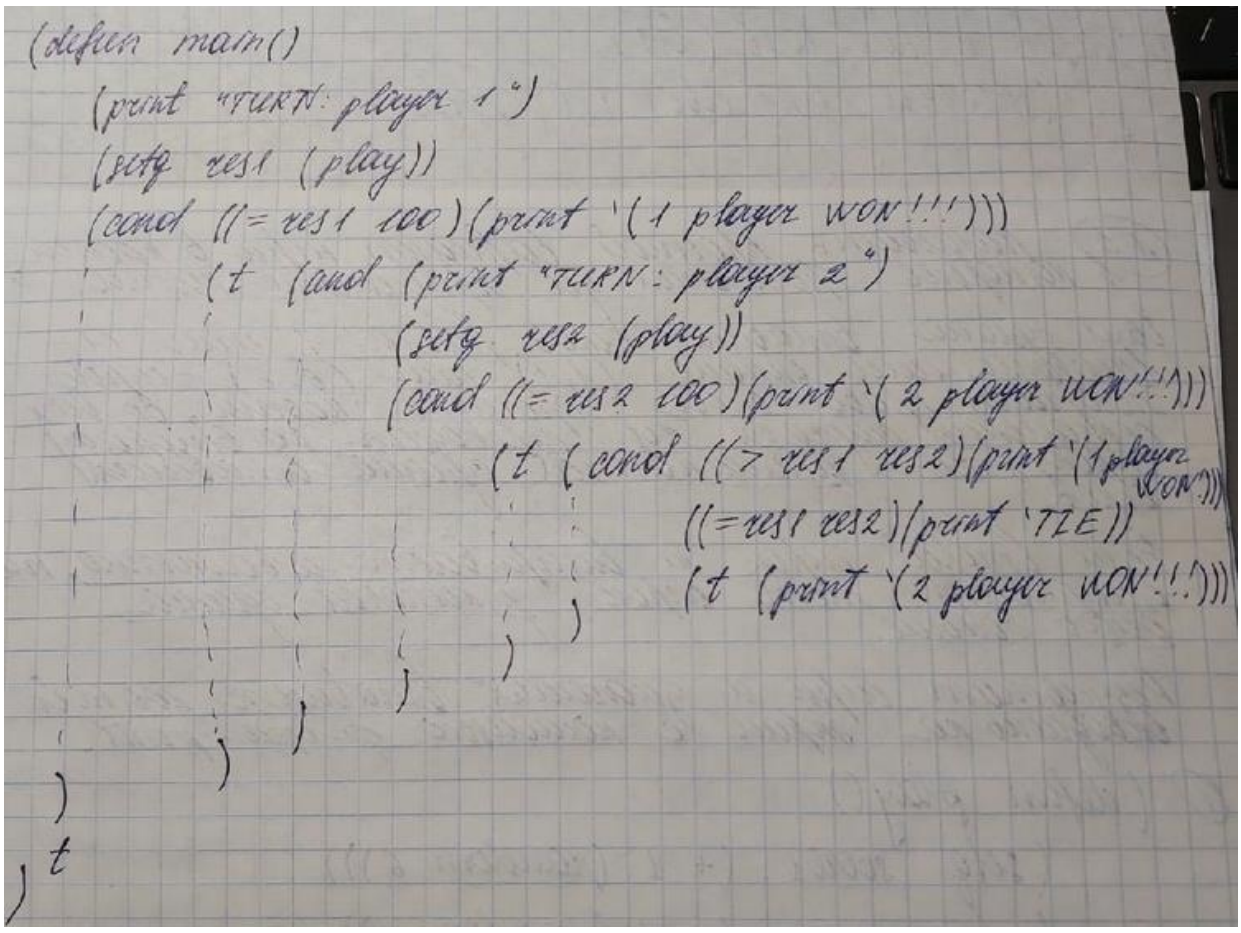
(№5) Написать простой вариант игры в кости, в которой бросаются две правильные кости.

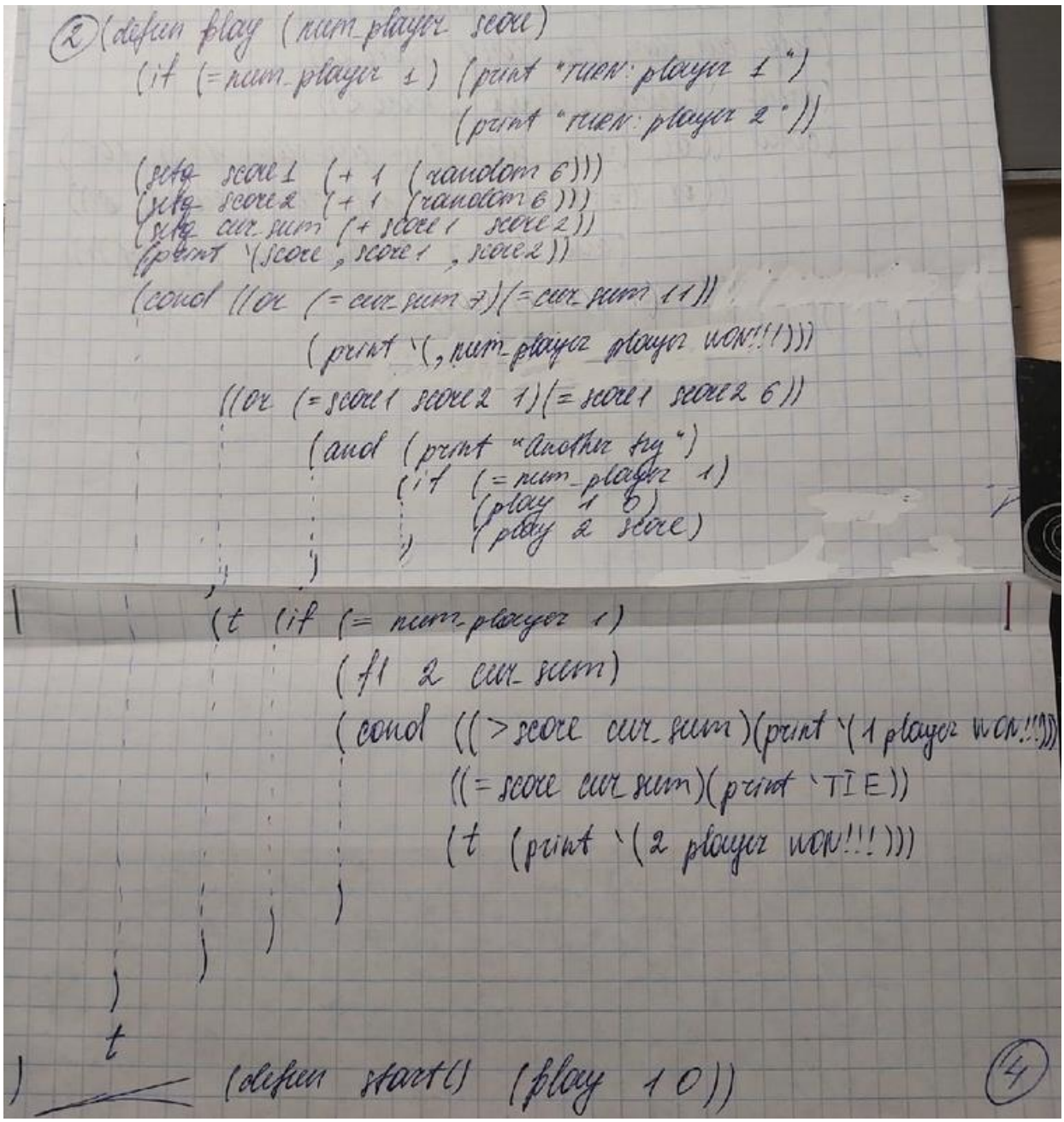
Если сумма выпавших очков равна 7 или 11 - выигрыш, или выпало (1,1) или (6,6) - игрок получает право снова бросить кости, во всех остальных случаях ход переходит ко второму игроку, но запоминается сумма выпавших очков.

Если второй игрок не выигрывает автоматически, то выигрывает тот игрок, у которого сумма очков больше.

Результат игры и значения выпавших костей выводить на экран с помощью ф-ции print.

(1)
```
(defun play()
    (setq score1 (+ 1 (random 6)))
    (setq score2 (+ 1 (random 6)))
    (setq cur_sum (+ score1 score2))
    (print '(score , score1 , score2))
    (cond ((or (= cur_sum 7)(= cur_sum 11)) 100)
          ((or (=score1 score2 1)(=score1 score2 6))
              (and (print "Another try ")(play)))
          (t cur_sum)
    )
)
```


(defun main()
(print "TURN: player 1")
(setq res1 (play))
(cond ((= res1 100)(print '(1 player WON!!!)))
(t (and (print "TURN: player 2")
(setq res2 (play))
(cond ((= res2 100)(print '(2 player WON!!!)))
(t (cond ((> res1 res2)(print '(1 player WON)))
((= res1 res2)(print 'TIE))
(t (print '(2 player WON!!!)))
)
)
)
)
)
t
)



```
(defun play (num_player score)
  (if (= num_player 1) (print "turn: player 1")
                       (print "turn: player 2"))
  (setq score1 (+ 1 (random 6)))
  (setq score2 (+ 1 (random 6)))
  (setq cur_sum (+ score1 score2))
  (print '(score , score1 , score2))
  (cond ((or (= cur_sum 7)(= cur_sum 11))
         (print '(, num_player player won!!!)))
        ((or (= score1 score2 1)(= score1 score2 6))
         (and (print "another try")
              (if (= num_player 1)
                  (play 1 0)
                  (play 2 score)
              )
         )
        (t (if (= num_player 1)
               (f1 2 cur_sum)
               (cond ((> score cur_sum)(print '(1 player won!!!)))
                     ((= score cur_sum)(print 'TIE))
                     (t (print '(2 player won!!!)))
               )
           )
        )
  )
  t
)

(defun start() (play 1 0))
```

(2)

(4)

Вопрос

① Структуроразрушающие и неразрушающие структуру списка функции

Функции делятся на:

- разрушающие структуру (если после окончания работы ф-ции утрачивается возможность работать с теми данными, которые были доступны до этого)
- неразрушающие

Некоторые примеры:

Разрушающие	Неразрушающие
• nconc	• append
• nreverse	• reverse
• rplaca / rplacd	• remove
• nsubst	• subst
	• union
	• intersection

② Отличие в работе ф-ций cons, list, append и в их результате

cons – принимает только 2 аргумента; создает только одну списковую ячейку, а перенаправляет указатели на значения

list – принимает произвольное количество аргументов; создает столько списковых ячеек, сколько аргументов, car-указатели этих ячеек устанавливаются на аргументы, с помощью cons указателей связываются списковые ячейки, последний эл-т в Nil

append – принимает произвольное количество аргументов, создает копию всех списков, которые подаются на вход, кроме последнего (копируются списковые ячейки)