Тема: **Функции высших порядков, λ-выражения**  
  
I. Написать решение задачи на функциональном языке, проверить его в интерпретаторе HFL, вызвав функцию с разными аргументами.  
II. Переписать решение на Scheme и Kotlin и проверить аналогичным образом.

Определение функции для обработки списков на каждом из 3 языков программирования оценивается в 5 баллов. Оценка снижается за каждую ошибку или перевода определения на 1 балл.

*Варианты заданий*

1. Написать функцию интеграл(f,a,b,n) для численного интегрирования заданной функции f на интервале от a до b (интервал разбивается на n отрезков).

интеграл(/\(x)x^2,1,5,100) → 41.331

2. Используя функции отобр и соединить,

определите функцию, вычисляющую декартово

произведение двух множеств.

декарт('(1 2 3),'(4 5)) → ((1 4) (1 5) (2 4) (2 5) (3 4) (3 5))

3. Написать функцию каждый(f,s), проверяющую, что каждый элемент

списка s удовлетворяет функции-предикату f.

каждый(atom,'(A B C)) → #T

каждый(atom,'(A (B C))) → #F

4. Написать функцию замена(a,b,c), которая возвращает функцию.

заменяющую в списке чисел все вхождения значений от a до b включительно на c.

(замена(2,4,5))('(1 2 3 0 4 6)) → (1 5 5 0 5 6)

5. Написать функцию какие(f,s), возвращающую индексы элементов

списка s, которые удовлетворяют функции-предикату f.

какие(atom,'(a (b c) d e)) → (1 3 4)

6. Написать функцию применить(f,x), которая возвращает список результатов применения функций из списка функций f к элементам из списка x:

(f1(x1) ... fk(xk)).

применить(cons(/\(z)z+1,cons(/\(z)z+2,cons(/\(z)z+3,'()))),'(7 2 3)) → (8 4 6)

7. Написать функцию максдлина(f,s), вычисляющую максимальную длина непрерывной подпоследовательности из элементов списка s, которые удовлетворяют функции-предикату f.

максдлина(atom,'((A) B (C) D E F)) → 3

8. Написать функцию существует(f,s), проверяющую, что существует элемент

списка s удовлетворяющий функции-предикату f.

существует(atom,'((A B) C)) → #T

существует(atom,'((A B) (C))) → #F

9. Написать функцию применить(f,x,y), которая применяет бинарную функцию f последовательно ко всем элементам списков x и y, и возвращает список результатов (f(x1,y1), f(x2,y2), ..., f(xn,yn)).

Предполагается, что списки x и y имеют одинаковую длину.

применить(`+,'(1 2 4),'(6 11 2)) → (7 13 6)

10. Написать функцию пары(f,x), которая находит все пары элементов из списка x, для которых бинарная функция-предикат f возвращает #T.

пары(`<,'(1 7 4)) → ((1 7) (1 4) (4 7))

11. Написать функцию колво(f,x), которая возвращает

количество элементов списка x, удовлетворяющих функции-предикату f.

колво(atom,'((A) B (C) D E F)) → 4

12. Написать функцию разделить(f,x), которая разделяет

список x на два списка, в первый список должны войти элементы,

удовлетворяющие функции-предикату f, во второй - не удовлетворяющие.

разделить(atom,'((A) B (C) D E F)) → ((B D E F) ((A) (C)))

13. Написать функцию первый(f,x), которая возращает первое

значение из списка x, для которого функция-предикат f возвращает #T, и его индекс.

первый(atom,'((A) B (C) D E F)) → (B 2)

14. Написать функцию заменить(f,x,a), которая заменяет

последнее вхождение значения, для которого функция-предикат f

возвращает #T, в списке x на значение a.

заменить(atom,'((A) B (C) D E (F)),'Z) → ((A) B (C) D Z (F))

15. Написать функцию удаление(a), которая возвращает функцию,

удаляющую все вхождения значений меньше a в списке чисел.

(удаление(3))('(1 2 3 0 4 2)) → (3 4)

16. Написать функцию применить(f,x), применяющую функцию f ко всем атомам сложного списка x, включая все подсписки.

применить(/\(z)z+1,'(1 ((2) 3) 4)) → (2 ((3) 4) 5)

17. Написать функцию найти(f,x), которая возвращает

длину и индекс первого элемента первой непрерывной

подпоследовательности из элементов списка x,

удовлетворящих функции-предикату f.

найти(atom,'((A) B D E (C) (F)) → (3 2)

18. Написать функцию существует(f,s), проверяющую, что существует два элемента из списка s, удовлетворяющие бинарной функции-предикату f.

существует(`=,'(1 2 3 1)) → #T

существует(`=,'(1 2 3 4)) → #F

19. Написать функцию пары(f,x,y), которая находит все пары элементов (xi,yj), для которых бинарная функция-предикат f возвращает #T.

пары(`=,'(1 7 4), '(2 4 6 7)) → ((7 7) (4 4))

20. Написать функцию заменить(f,x,a), которая заменяет

все значения в списке чисел x, для которых функция-предикат f

возвращает #T, на значение a.

заменить(atom,'((A) B (C) D E (F)),'Z) → ((A) Z (C) Z Z (F))

21. Написать функцию замена(a,c), которая возвращает функцию

заменяющую все вхождения значений, кратных a, на c в списке чисел.

(замена(2,5))('(1 2 3 0 4 10)) → (1 5 3 5 5 5)

22. Пусть функция f является функцией принадлежности к множеству S

т.е. f(x)=если x в S то #T иначе #F

Написать функцию добавить(y,f) которая возращает

функцию принадлежности для множества S'= S &cup; {x}

less10(x)=x<10;

f2010=добавить(20,less10);

f2010(5) → #T

f2010(15) → #F

f2010(20) → #T

23. Написать функцию удалить\_если\_не(f,s), возращающую новый список из элементов списка s, удовлетворяющих функции-предикату f.

удалить\_если\_не(atom,'(A B (C) D)) → (A B D)

24. Написать функцию проверить(f,x,y), где f - функция, a x и y - списки, которая позволяет проверить соответствие функции точкам, вычисляя корень

из суммы квадратов f(xi)-yi

проверить(/\(z)z^2,'(1 2 3),'(1 5 8)) → 1.41421

25. Написать функцию линприменить(f,s), которая выполняет линеаризационное применение функции f к списку s.

линприменить(`+,'((1 2) ((3) 5))) → 11

линприменить(соединить,'((A B) (((C)) D) E (F G))) → (A B C D E G)

Функцию соединить необходимо доопределить для атомов.