

Задачи начального уровня по языку Lisp

1. Глубина списка (Взята!)

Постройте функцию, которая заданному списку определит максимальную вложенность. Глубина элемента 0.

Пример

```
(d '(a (b) (c (d))))
```

2

2. Мощность списка (Взята!)

Постройте функцию, которая заданному списку определит суммарное количество атомарных элементов.

Пример

```
(d '(a (b) (c (d))))
```

4

3. Список атомов (Взята!)

Постройте функцию, которая заданному списку выдаст список входящих в него и в его подсписки атомов, без повторений.

Пример

```
(d '(1 (3) (2 (1))))
```

```
( 1 3 2)
```

4. Леанирезация списка (Взята!)

Постройте функцию, которая заданному списку создаст список атомарных элементов в том же порядке.

Пример

```
(d '(a (b) (c (d))))
(a b c d)
```

5. Максимальный список (Взята!)

Постройте функцию, которая заданному списку выдаст подсписок в котором больше всего элементов.

Пример

```
(d '(a (b) (c (d))))
(c (d))
```

6. Зеркальный список (Взята!)

Постройте функцию, которая перевернёт список и все его подсписки. всего элементов.

Пример

```
(d '(a (b) (c (d))))
(((d) c) (b) a)
```

7. Максимум списка (Взята!)

Постройте функцию, которая найдет среди всех чисел списка, учитывая подсписки, максимальное и все его подсписки. всего элементов.

Пример

```
(d '(1 (2) (x (3))))
3
```

8. НОД списка (Взята!)

Если список чисел (без подписка). Постройте функцию, которая найдет Наибольший общий делитель этих чисел.

Пример

```
(d '( 10 100 1000 10000))
10
```

9. Путь (Взята!)

Дан список пар названий комнат и ещё две комнаты. Пара (a b) обозначает, что можно попасть из комнаты a в комнату b. Построить функцию, которая определит есть ли путь из первой комнаты во вторую и выдаст его.

Пример

```
(d '( (312-1 312) (312-2 312) (312-3 312) (312-4 312) (301 x)
(312 x) (x 314) (x x) ) 312-1 314 )
(312-1 312 x 314)
```

10. Достижимость (Взята!)

Дан список пар названий комнат и ещё комната. Пара (a b) обозначает, что можно попасть из комнаты a в комнату b. Выдать список комнат в которые можно добраться из данной.

Пример

```
(d '( (312-1 312) (312-2 312) (312-3 312) (312-4 312) (301 x)
(312 x) (x 314) (x x) ) x )
(x 314)
```

11. Циклический сдвиг вправо (Взята!)

В заданном списке осуществить циклический сдвиг вправо на 1.

Пример

```
(d '(a b c d))
(d a b c)
```

12. Циклический сдвиг влево (Взята!)

В заданном списке осуществить циклический сдвиг влево на 1.

Пример

(d '(a b c d))

(b c d a)

13. Циклический сдвиг вправо-2 (Взята!)

В заданном списке осуществить циклический сдвиг вправо на 1.

Пример

(d '(a b c d))

(c d a b)

14. Циклический сдвиг влево-2 (Взята!)

В заданном списке осуществить циклический сдвиг влево на 1.

Пример

(d '(a b c d))

(c d a b)

15. Поиск одинаковых (Взята!)

Выдать список элементов у которых есть копии в этом списке.

Пример

(d '(a b c d a b a c a a))

(a b c)

16. Двойной список (Взята!)

Если список есть запись двух одинаковых списков, то выдать «Да», иначе выдать «Нет».

Пример

(d '(a b c a b c))

Да

17. Сокращаемый список (Взята!)

Если список есть запись нескольких одинаковых списков, то выдать «Да», иначе выдать «Нет».

Пример

```
(d '(a b a b a b))
```

Да

18. Список одного (Взята!)

Если для построения списка и его подсписков был использован только 1 элемент, выдать его, иначе выдать «Нет».

Пример

```
(d '(a (a) (a a a) (((a))) )
```

a

19. Список разных (Взята!)

Если для построения списка и его подсписков были использованы разные элементы, выдать «Да», иначе один из повторяющихся элементов.

Пример

```
(d '(a (1) (2 b a) (((c))) )
```

a

20. Цикл

Дан список пар названий комнат и ещё две комнаты. Пара (a b) обозначает, что можно попасть из комнаты a в комнату b. Построить функцию, которая определит можно ли ходить по каким-то комнатам по-кругу и выдаст список этих комнат.

Пример

```
(d '( (312-1 312) (312-2 312) (312-3 312) (312-4 312) (301 x)
(312 x) (x 314) (x x) ) (314 301) )
```

(301 312 x 314)

21. Производная (Взята!)

В качестве параметра создаваемая Вами функция получает функцию действительного переменного, запрограммированную при помощи лишь сложения и умножения. Посчитать ее производную. Алгебраические упрощения делать не обязательно.

Внимание! При тестировании задачи функция может быть задана некорректно. Ошибку в ее задании нужно вылавливать.

Пример

```
(d '(lambda (a)(+ (* a a) x)) )
(lambda (a)(* 2 a) )
```

22. Интерполяция (Взята!)

Задан список аргументов и значений не более чем из 10 пар. Выдать функцию, проходящую через эти точки.

Пример

```
(d '( (1.0 1.0) (2.0 2.0) (3.0 3.0) (4.0 4.0))) )
(lambda (a)( a) )
```

23. Интеграл (Взята!)

В качестве параметра создаваемая Вами функция получает функцию действительного переменного, запрограммированную при помощи лишь сложения и умножения. Посчитать ее интеграл. Алгебраические упрощения делать не обязательно.

Внимание! При тестировании задачи функция может быть задана некорректно. Ошибку в ее задании нужно вылавливать.

Пример

```
(d '(lambda (a)(+ (* a a) x)) )
(lambda (a)(+ (* 0.33333 (* a (* a a))) c) )
```

24. Логические функции

В качестве параметра создаваемая Вами функция получает логическую функцию нескольких переменных, запрограммированную при помощи лишь логических операций: конъюнкции, дизъюнкции и отрицания, и список значений некоторых из ее параметров. Выдать остаточную

функцию, получающуюся после подстановки значений в исходную функцию и ее упрощения.

Внимание! При тестировании задачи функция может быть задана некорректно. Ошибку в ее задании нужно вылавливать.

Пример

```
(d '(lambda (a b c)(and (or a b) (or (not a) (not c)))) ((a
t) (b f)) )
(lambda (c)(not c))
```

25. Многочлен

В качестве параметра создаваемая Вами функция получает многочлен нескольких переменных, задаваемый при помощи лишь операций: + и *, причем + не будет встречаться внутри * и в каждом слагаемом не более одного постоянного коэффициента. и список значений некоторых из ее параметров. Выдать остаточную функцию, получающуюся после подстановки значений в исходную функцию и ее упрощения. Коэффициенты и параметры — целые числа.

Внимание! При тестировании задачи функция может быть задана некорректно. Ошибку в ее задании нужно вылавливать.

Пример

```
(d '(lambda (a b c)((+ (* 2 a b c) (+ (* 2 c b a) -3)) ((a
1) (b -1)) )
(lambda (c)(+ (* -4 c) -3))
```

26. Собери многочлен

В качестве параметра создаваемая Вами функция получает список списков сомножителей. Функцию вычисляющий многочлен, который получится, если сложить все элементы данного списка (каждый элемент — произведение). Коэффициенты и параметры — целые числа.

Внимание! При тестировании задачи функция может быть задана некорректно. Ошибку в ее задании нужно вылавливать.

Пример

```
(d '((2 a b c) (c b) (a 10 -10) (b c) (-3))
(lambda (a b c)(+ (* 2 a b c) (* 2 b c) -3) )
```

27. Красивый многочлен (Взята!)

В результате дедуктивного синтеза был получен многочлен, в котором умножения и сложения только двухместные и возможна куча подобных членов. Но все умножения внутри сложения. Коэффициенты и параметры — целые числа.

Привести подобные и записать с минимальным количеством операций

Внимание! При тестировании задачи функция может быть задана некорректно. Ошибку в ее задании нужно вылавливать.

Пример

```
(d '(lambda (a b c)((+ (* 2 (* (* a b) c) (+ (* (* 1 c) (*(1 c)) (* b a)) (+ -1 -2)) )
(lambda (a b c)(+ (* 2 a b c) (* 2 b c) -3) )
```

28. Минимум

Задан список элементов, первый из которых функция минимума для двух элементов. Остальной список — элементы, среди которых нужно найти минимальный.

Пример

```
(d '(lambda (x y)((if (> x y) x y) a b) )
c
```