

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Алтайский государственный технический университет
им. И.И. Ползунова»

А.Ю. Андреева

Методические указания к лабораторному практикуму по дисциплине

«Функциональное и логическое программирование»

для бакалавров направления «Программная инженерия»

Изд-во АлтГТУ
Барнаул 2020

УДК.

Андреева А.Ю.. Методические указания к лабораторному практикуму по курсу **«Функциональное и логическое программирование»** для бакалавров направления «Программная инженерия»/А. Ю. Андреева; Алт. гос. техн. ун-т им. И. И. Ползунова. – Барнаул: Изд – во АлтГТУ, 2020. – 18 с.

Методические указания представляют собой руководство к лабораторным занятиям по курсу Функциональное и логическое программирование

Рекомендовано бакалаврам дневной формы обучения направления ПИ

Рассмотрено и одобрено
на заседании кафедры

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ №1

ИСПОЛЬЗОВАНИЕ ВСТРОЕННЫХ ФУНКЦИЙ ЛИСПА

Необходимо выполнить по одному заданию из каждого упражнения (вариант взять у преподавателя).

Задания выполняются в среде cLisp (или другой). Результаты привести в отчете. В случае ошибок интерпритации – привести пояснения.

Быть готовым пояснить любой вопрос по тексту кода.

Упражнение 1

Записать следующие выражения в виде s-выражений Лиспа:

- | | |
|--|---------------------------------|
| 1. $(2+3*6)/4/(5*(2+3))$ | 16. $(36+8)/(7+5)*4$ |
| 2. $(6+3*4)*(6+2)/6+6$ | 17. $(-3.6+2.4)*5/(25/5)$ |
| 3. $2+4.3/6.8-4/7.2+3*5$ | 18. $96/(16-(18*0))+7$ |
| 4. $(7.2-3.6)/(4*5-4.8)+3.6*5$ | 19. $25*25/(5*5+10*10)$ |
| 5. $(3.8-2.4)/3.8+4.2*3/6.8+4$ | 20. $(7.3-6.8-(-3.4))*(92+3.4)$ |
| 6. $(5-3)/(4.8-6.2)+3.4/(7-2)+3$ | 21. $(2*2+3*3)-(6.4-4.4)$ |
| 7. $7-6/4+3*4/(7-2)+3*4$ | 22. $-35+78*87/78$ |
| 8. $(3-4.8*2)/(6*4-3.8)-(4.2-2)/(4-3)$ | 23. $(5.5+6.6-7.7*8.8)/9.9$ |
| 9. $(-3+4.2*6)/(7.8-3)+4.8/(3.4-2)$ | 24. $56+65/56-25*52/4$ |
| 10. $(3.4-3.2)/(-1.6)-3.8/(2-6)+4.8$ | 25. $77+88-7*(7*8/4)$ |
| 11. $(2.5+7)/8.1*(6.5+4.3)$ | 26. $92/4*8+89-72$ |
| 12. $(1.2-6.8)/3*5+(2-8)$ | 27. $72/9+48*(6/3)$ |
| 13. $(7.8-3)*(5+4)/3-1.8$ | 28. $2.5+4.6*7.4-4.8$ |
| 14. $8.3*(1.1-1.0)+7.6/3.8$ | 29. $3.9+9.3/3.9*2.5$ |
| 15. $6+3/7*8+(3+5)/2.3$ | 30. $(88-8.8)/((77+7.7)*4.4)$ |

Упражнение 2

Определить, как будут выполнены интерпретатором Лиспа следующие последовательности s-выражений, т.е. что появится на выходе и какие значения получат символьные атомы в результате присваиваний

1. $(\text{SETQ } A \ 3)$
 $(\text{SETQ } B \ A)$

- (SET (SETQ B 'A) 'B)
2. (SETQ B 3)
(SETQ A 'B)
(SETQ '(SET A 4) B)
3. (SETQ A 'C)
(SETQ B 2)
(SETQ (SET A B) 'C)
4. (SET 'B 4)
(SET A B)
(SET '(SETQ A 'B) 2)
5. (SETQ A 'C)
(SETQ B 'A)
(SETQ B (SETQ B 'C))
6. (SETQ B 2)
(SET '(SET B 'C) 'Z)
7. (SETQ 'A 4)
(SETQ 'B 'A)
(SETQ (SET B A) '3)
8. (SETQ B 'A)
(SETQ C 'B)
(SET (SET B C) B)
9. (SETQ B 'A)
(SET B (SET B '2))
(SETQ (SET 'A B) 'C)
10. (SETQ A 2)
(SET 'C 'B)
(SETQ (SET 'C 'B) A)
11. (SETQ A 'C)
(SET (SET A 'B) (SET 'C B))
12. (SETQ (SETQ A 'B) 'C)
(SET (SET A B) (SET B 'A))

13. (SETQ B 'C)
(SET B 'A)
(SET B (SETQ A 'C))
14. (SETQ B 'A)
(SET (SETQ C B) (SETQ 'A C))
15. (SETQ C 'A)
(SET C 'B)
(SET C (SETQ 'B '2))

Упражнение 3

Записать последовательность функций CAR и CDR, выделяющую из приведенных списков атом A.

1. (1 (2 (A) 3) 4)
2. ((1) (2 A) 3)
3. ((1 (2 (A))))
4. (1 2 (3 A))
5. (1 (2) 3 (A))
6. (1 2 ((3) A))
7. (1 2 (3 (A 4)))
8. ((1 2) (3 A) 4)
9. ((1 2) ((3 A) 4))
10. ((1) (2) (A))
11. ((1 2) (3 A))
12. ((1 2) (3 (A)))
13. (1 (2 A))
14. (1 ((2 A) 3))
15. ((A) (1 2))
16. (3 ((A) 1 2) 3)
17. ((1 (A)) 2)
18. ((1 A) 2)
19. (1 (2) (A 3))
20. (1 (A 2) 3)
21. ((A 1) 2)
22. (1 2 ((A) 3))
23. ((1 (2) A) 3)
24. ((1 2) (A (3 4)))
25. ((A (1)) 2)
26. (1 ((2 (A)) 3))
27. ((1 (A)) 2 3)
28. (1 2 ((A) 3))
29. ((1 2) 3 (4 A))
30. (1 (2 ((A) 3))

Упражнение 4

Каков результат интерпретации следующих выражений?

1. (CONS NIL '(A B C))
2. (CONS '(NIL) '(NIL))
3. (CONS NIL NIL)
4. (APPEND '(A) NIL)
5. (LIST (SETQ A 'B) 2 '(A))
6. (LIST 'A (SETQ A '(B)) 3 A)

7. (CAR (SET 'A '(B C D)))
8. (CAR (CDR '(A (B C))))
9. (CDR (CDR '(CDR '(A B C))))
10. (CAR '((A) (B C)))
11. (CDR (LIST 'A 'B NIL))
12. (CAR ''A)
13. (LIST (LIST 'A 'B) '(CAR '(C D)))
14. (LIST NIL '(A))
15. (LIST NIL NIL)
16. (APPEND (CAR '((A))) '(B (C)))
17. (APPEND (CDDR '(A B)) '(C D))
18. (APPEND NIL NIL)
19. (LIST (CDR '(A B C)) (CAR '(A B C)))
20. (CONS '(A) '(B))
21. (CONS 'A (LIST (+ (/ 6 2) 5)))
22. (CONS 'A '(+ (/ 6 2) 5))
23. (APPEND (CDR '((A) B C)) (CAR '((A) B C)))

Упражнение 5

В данном упражнении требуется получить списки справа от стрелки с помощью суперпозиции функций LIST, APPEND, CONS, CAR, CDR. Аргументами функций могут быть лишь s-выражения, указанные слева от стрелки.

Пример: (A B); C; D \Rightarrow (A B (C D))

Решение: (APPEND '(A B) (LIST (LIST 'C 'D)))

- | | | |
|-----------------------|---------------|------------------|
| 1. (A B) C (D) | \rightarrow | ((A B) (C) D) |
| 2. (A B) (C); (D) | \rightarrow | ((A B) C D) |
| 3. (A B) (C D) | \rightarrow | (A (B C) D) |
| 4. (A); (B); (C); (D) | \rightarrow | ((A B) (C D)) |
| 5. (A); B; C; D | \rightarrow | (A B (C) (D)) |
| 6. (A); B; C; D | \rightarrow | (((A) (B)) C D) |
| 7. A; B; (C D) | \rightarrow | (A (B C) D) |
| 8. A; (B C); D | \rightarrow | ((A) (B) C (D)) |
| 9. (A); (B); (C D) | \rightarrow | (A (B C) D) |
| 10. (A B); (C (D)) | \rightarrow | (A (B (C)) D) |
| 11. (A B); (C D) | \rightarrow | ((A) B C (D)) |
| 12. A; B; (C D) | \rightarrow | ((A B) C D) |

- | | | |
|---------------------|---|---------------|
| 13. A; (B C D) | → | ((A) B (C D)) |
| 14. A; (B) ; C; (D) | → | ((A) B (C D)) |
| 15. A; (B C); (D) | → | ((A B) (C) D) |
| 16. (A); B; (C D) | → | (A (B) (C) D) |
| 17. A; (B C); (D) | → | (A (B) (C D)) |
| 18. (A); (B C); D | → | (A (B (C) D)) |

Упражнение 6

Какое из выражений дает значение T, а какое NIL?

1. (ATOM '(CAR '(A B)))
2. (EQUAL '(A B) (CONS '(A) '(B)))
3. (EQ '(A B) (CONS 'A '(B)))
4. (EQ 1234567890 1234567890)
5. (EQUAL '1234567890 1234567890)
6. (EQ 5 20/4)
7. (EQ 1/5 0.2)
8. (ATOM (* 2 (+ 2 3)))
9. (NULL (NULL T))
10. (EQUAL (ATOM NIL) (CAAR '((T))))
11. (ATOM (SETQ A 'B))
12. (NULL (LISTP (CAR '(A B))))
13. (NUMBERP '(+ 2 3))
14. (LISTP (CAR '(A)))
15. (LISTP (EQ 'A 'B))
16. (ATOM (CAR '(SETQ A 'B)))
17. (NULL (LISTP NIL))
18. (NUMBERP '(SETQ A 4))
19. (ATOM '(CAR '(A)))
20. (ATOM `(LISTP `(A)))
21. (ATOM (LISTP `(A)))
22. (CONSP (CONS 'A 'B))
23. (CONSP (CAR `(A B)))
24. (CONSP (CDR `(A)))
25. (EQ (SETQ A 'B) A)
26. (EQ (SETQ A 'B) (CAR '(A X)))
27. (EQ (SETQ A 'B) (SETQ A 'C))
28. (EQ (SETQ A 'B) (SET 'A A))
29. (ATOM (CAR '(CAR '((A) B))))
30. (CAR (LIST NIL))

Упражнение 7

Чему равны значения следующих выражений?

1. (AND (CDDR ' (A B)) 'A 2)
2. (OR (CAR ' (NIL 2 'A)) (ATOM ' (A B)))
3. (OR (SET 'A NIL) (AND (NULL A) (CDR ' (A))))
4. (AND (> 5 3) (< (SETQ A 1) 0))
5. (OR (SETQ A NIL) (CAR ' (A B)))
6. (AND (SETQ A '5) (SETQ 'C (= A 4)))
7. (OR ' (A 4) (SETQ C (A 4)))
8. (OR (SETQ C (= 4 8)) (CAR ' (NIL C)))
9. (AND (OR (SETQ A 5) (SETQ B A)) (SETQ C B))
10. (OR (ATOM (< 5 3)) (SETQ C '8))
11. (OR (SETQ A 'B) (SETQ 'B C))
12. (AND (SETQ B 'A) (EQ A B))
13. (SET (OR (SETQ A NIL) (SET 'A 'B)) (AND (CAR ' (A B)) A))
14. (AND (OR ' (SETQ A NIL) NIL) (OR NIL A))
15. (AND (EQUAL (SETQ A NIL) NIL) (EQ (OR NIL A 2) 2))
16. (AND (NULL (SETQ A NIL)) (EQ A (ATOM ' (A B))))
17. (OR (APPEND NIL NIL) (CONS NIL NIL) 2)
18. (AND (LIST NIL NIL) (CAR (LIST NIL)) (CONS NIL NIL))
19. (SETQ (OR (SETQ A NIL) (SETQ A 'B) (SETQ B 'C)) (SET B 2))
20. (OR (CDR ' (NIL)) (CAR ' (NIL)) (LIST NIL))

Упражнение 8

Записать определения следующих функций, используя функции проверки условия IF и COND (т.е. 2 варианта!):

1. Функция, вычисляющая выражение $1/a + 1/b$ для заданных чисел a и b. В случае, если какое-либо из чисел равно нулю, функция должна давать нуль.
2. Функция, вычисляющая максимум из трех чисел a, b, c.
3. Функция, вычисляющая квадрат наименьшего из двух чисел a и b.
4. Функция, имеющая один аргумент-список и возвращающая квадрат третьего элемента этого списка, если он является числом, голову третьего элемента, если он является списком, иначе сам третий элемент.
5. Функция, имеющая один аргумент-список и возвращающая сумму 4-го и 5-го элемента этого списка, если оба этих элемента являются числами, иначе 1-й элемент списка.
6. Функция, имеющая один аргумент-список и возвращающая квадрат первого элемента этого списка, если этот элемент является числом, и 3-й элемент списка в противном случае.

7. Функция, имеющая один аргумент-список и возвращающая список из 2-го и 4-го элемента, если они оба являются атомами, иначе квадрат 1-го элемента, если этот элемент - число, и NIL в противном случае.

8. Функция, имеющая два аргумента - числовых списка и возвращающая тот список, голова которого меньше.

9. Функция, аргументами которой являются координаты двух точек $(x_1 \ y_1)$ $(x_2 \ y_2)$, а результатом - расстояние между точками

10. Функция, имеющая три аргумента и возвращающая их сумму, если все они числа, и список, состоящих из них, в противном случае.

11. Функция, аргументом которой является числовой список, а результатом максимум из квадрата головы и суммы второго и третьего элементов.

12. Функция, аргументом которой является некоторый список, а результатом голова этого списка, если в списке больше трех элементов, иначе второй элемент списка.

13. Аргументом функции является список из трех неравных чисел. Функция должна вернуть число, которое не является максимумом и минимумом.

14. Функция имеет 2 числовых аргумента a и b описывающих уравнение $ax+b=0$. Значением функции должен быть корень уравнения.

15. Аргументом функции является список из трех чисел. Функция должна вернуть список из тех же чисел, расположенных в порядке возрастания.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ №2

МЕТОДЫ ХВОСТОВОЙ РЕКУРСИИ ДЛЯ ОБРАБОТКИ СПИСКОВ

Задание: Реализовать обработку списков с использованием механизма общей рекурсии.

Задания выполняются в среде cLisp (или другой). Код программ и тестовые примеры привести в отчете.

Быть готовым пояснить любой вопрос по тексту кода.

Варианты:

1. Написать функцию вычисления m^n .
2. Для произвольного списка определить, является ли последний элемент списка списком или атомом.
3. Получить список, состоящий из поэлементной суммы двух списков.
(2 4 1 5 6) (3 4 1 2 2) \rightarrow (5 8 2 7 8)
4. Написать функцию-предикат, проверяющую, принадлежит ли данный элемент заданному списку:
(MEMBER 'A '(S D A H)) \rightarrow T
(MEMBER 'D '(H K (D) G)) \rightarrow NIL
5. Для произвольного списка построить список той же длины, состоящий из атомов T и NIL в зависимости от того, является ли соответствующий элемент исходного списка списком или атомом соответственно.
6. Написать функцию, которая по заданному s-выражению A и числу N строит список вида ((((((A)))))), где количество скобок с каждой стороны от выражения равно N.
7. Написать функцию, преобразующую список к виду
(A B C) \rightarrow (A (B (C)))
8. Определить функцию, разбивающую список на пары:
(A B C D) \rightarrow ((A B) (C D))
9. Написать функцию, которая, чередуя элементы двух списков, строит новый список:
(A B C) (1 2 3) \rightarrow (A 1 B 2 C 3)
10. Определить функцию, которая строит список (1 3 5 7 ... $2*N-1$) для заданного N.
11. Для заданного числового списка построить новый список, в котором

вначале находятся все отрицательные элементы, а затем неотрицательные.

12. Два множества представлены списками. Найти пересечение множеств.
13. Два множества представлены списками. Найти вычитание множеств.
14. Два множества представлены списками. Найти объединение множеств.
15. Дан список из положительных чисел, построить список, элементами которого являются количество единиц в исходном списке, количество двоек и т.д. до максимального значения числа в исходном списке:
 $(2\ 1\ 1\ 3\ 1\ 3\ 5) \longrightarrow (3\ 1\ 2\ 0\ 1)$
16. Из данного списка удалить все элементы, которые являются списками длиной более K.
17. Выполнить попарные перестановки элементов списка
 $(A\ B\ D\ F\ B\ A\ C) \longrightarrow (B\ A\ F\ D\ A\ B\ C)$
18. Дан список, число и произвольное s-выражение. Поставить s-выражение в список перед элементом, номер которого равен числу:
 $(A\ B\ C\ D),\ T,\ 3 \longrightarrow (A\ B\ T\ C\ D)$
19. Написать функцию, превращающую список в множество, т.е. удаляющую все повторяющиеся элементы:
 $(A\ B\ A\ B\ B\ C\ A) \longrightarrow (A\ B\ C)$
20. Написать функцию, проверяющую, является ли список множеством, т.е. каждый элемент в нем должен встречаться по одному разу.
21. Проверить, состоят ли два списка из одних и тех же элементов.
22. Дан список и число N. Разбить список на подсписки длины N:
 $(A\ B\ (C\ D)\ 1\ 2\ NIL)\ 4 \longrightarrow ((A\ B\ (C\ D)\ 1)\ (2\ NIL))$
23. Разложить число на простые множители.—>
 $30 \longrightarrow (2\ 3\ 5)$
24. Дан одноуровневый список, состоящий из чисел. Построить список из последовательных сумм элементов.
 $(A_1, A_2, A_3, \dots, A_N) \longrightarrow (A_1, A_1+A_2, A_1+A_2+A_3, \dots, A_1+A_2+\dots+A_N)$
25. Для двухуровневого списка вернуть подсписок максимальной длины.
26. Для одноуровневого списка удалить элемент, стоящий следом за

заданным

$(A B A B B C A), A \rightarrow (A A B C A)$

27. Написать функцию, выполняющую над числовым одноуровневым списком следующую операцию:

$(A B C \dots) \rightarrow A^B C \dots$

Причем последовательное возведение в степень выполняется справа налево.

28. Написать функцию, которая для заданных произвольного списка и числового строит список вида:

$(a c g) (2 3 5) \rightarrow (((a)) (((c))) (((((g)))))))$

29. Написать функцию, вычисляющую произведение всех элементов одноуровневого числового списка в предположении, что отсутствует операция умножения и можно использовать лишь сложение.

30. Из произвольного списка и числового списка построить новый список:

$(A B C) (1 2 3) \rightarrow ((A) (B B) (C C C))$

31. Даны два множества, представленные списками. Проверить, является ли первое множество подмножеством второго.

32. Построить список, определяющий сколько раз встречается каждый элемент в списке: $(A B A C B C A B D) \rightarrow ((A 3) (B 3) (C 2) (D 1))$

33. Даны два числовых списка. Построить третий список, количество элементов которого равно длине второго списка и каждый элемент равен количеству элементов первого списка, меньших соответствующего элемента второго списка:

$(1 7 3 0 2) (3 7 10) \rightarrow (3 4 5)$

34. Написать функцию, удаляющую из списка каждый K-й элемент.

35. Разбить список на подсписки длины 1, 2, 3...

36. Дан список и числа N и M. Вырезать из списка подсписок от элемента N до элемента M:

$(A B C D E) 2 4 \rightarrow (B C D)$

37. Даны два множества, представленные списками. Определить функцию, строящую декартово произведение множеств:

$(A B C) (1 2) \rightarrow ((A 1) (A 2) (B 1) (B 2) (C 1) (C 2))$

38. Дан список из произвольных элементов. Найти наиболее часто встречающийся элемент.

39. Написать функцию, вычисляющую множество всех подмножеств

данного множества:

$(A\ B\ C) \longrightarrow (NIL\ (A)\ (B)\ (C)\ (A\ B)\ (A\ C)\ (B\ C)\ (A\ B\ C))$

40. Дано множество A и список из подмножеств данного множества B . Разбить A на два подмножества так, чтобы ни одно из подмножеств B не содержалось целиком в каждом из двух подмножеств. Например:

$(A\ B\ C\ D\ E)\ ((A\ B)\ (A\ C\ D)\ (D\ E)) \longrightarrow ((A\ C\ E)\ (B\ D))$

41. Дано натуральное число N . Построить список из подписков натуральных чисел, в сумме дающих N . Пример:

$3 \longrightarrow ((1\ 2)\ (2\ 1)\ (1\ 1\ 1))$

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ №3

МЕТОДЫ ОБЩЕЙ РЕКУРСИИ ДЛЯ ОБРАБОТКИ СПИСКОВ

Задание: Реализовать обработку списков с использованием механизма общей рекурсии.

Задания выполняются в среде cLisp (или другой). Код программ и тестовые примеры привести в отчете.

Быть готовым пояснить любой вопрос по тексту кода.

Варианты:

1. Написать функцию-предикат, проверяющую, содержится ли данный атом в списке:
 $(\text{MEMQ 'A '(S W F (A S))}) \rightarrow T$
 $(\text{MEMQ 'U '(G J (J))}) \rightarrow \text{NIL}$
2. Написать функцию, которая из заданного списка строит одноуровневый список:
 $(A (B (C 1) D)) \rightarrow (A B C 1 D)$
3. Для произвольного списка вычислить количество чисел на заданной глубине
 $((1 A) ((3 4) (D C)) (2) (A)), 2 \rightarrow 2$
4. Найти максимальный уровень вложенности заданного списка.
5. Найти последний отличный от NIL атом в заданном списке:
 $(A B (S 1) ((C)) \text{NIL}) \rightarrow C$
6. Дан произвольный список из числовых атомов. Найти минимальный атом в списке.
7. Для произвольного списка найти глубину, на которой находится первое вхождение заданного атома
 $((1 A) ((3 4) (D C)) (2) (A)), D \rightarrow 2$
8. Найти первый отличный от NIL атом в произвольном списке:
 $((()) A) B C \rightarrow A$
9. Даны два списка одинаковой структуры. Построить список такой же

структуры, состоящий из пар элементов.

$((A\ B(B)C)D) , ((1\ C\ (2)3)4) \longrightarrow (((A\ 1)\ (B\ C)((B\ 2))(C\ 3))(D\ 4))$

10. Дан список произвольной структуры, в который входят как числа, так и символьные атомы. Написать функцию, которая возвращает список из двух чисел, первое из них равно сумме чисел исходного списка, а второе их количеству.

$((1\ A)(2\ (3)\ B\ (4))) \longrightarrow (10\ 4)$

11. Дан список произвольной структуры. Каждый числовой атом списка необходимо увеличить на единицу.

$((A\ (1)2)B\ 3) \longrightarrow ((A\ (2)3)B\ 4)$

12. Найти самый глубоко расположенный атом в произвольном списке. Если таких несколько, вывести первый из них.

$(A\ (B\ C)(D\ (E)F)) \longrightarrow E$

13. Для произвольного списка вычислить сколько раз встречается заданный атом

$((1\ A)\ ((3\ 4)\ (D\ C))\ (2)\ (A)), A \longrightarrow 2$

14. Имеется произвольный список, состоящий из числовых атомов. Преобразовать список по следующему правилу: если элементы некоторого подсписка являются числами, заменить подсписок суммой его элементов.

$(1\ (2\ (3\ 4\ 5)\ 6)(7\ 8\ 9)) \longrightarrow (1(2\ (12)\ 6)(24))$

15. Дан список произвольной структуры. Каждый подсписок, состоящий из одного атома, заменить на сам атом.

$(A\ ((B)C)((D))) \longrightarrow (A(B\ C)D)$

16. Написать функцию, проверяющую, что каждый атом произвольного списка является числом.

17. Определить, сколько атомов находится на заданном уровне вложенности. Элемент может быть как атомом, так и списком.

$(A\ B\ (B\ A\ (C))\ A\ C\ (A)), 2 \longrightarrow 3$

18. Проверить, состоят ли два списка из одних и тех же атомов.

19. Удалить все числовые атомы в произвольном списке.

20. Имеется список произвольной структуры, но все одноуровневые списки в нем содержат ровно два элемента. Необходимо преобразовать список, удалив второй элемент.

$((1\ 2)\ ((3\ 4)\ (5\ 6))\ (((7\ 8)))) \longrightarrow ((1)((3)(5))(((7))))$

21. Дан список произвольной структуры, состоящий из чисел. Каждый атом увеличить на его уровень вложенности.

$$(1 (2 3((4)5))) \longrightarrow (2(4 5((8)8)))$$

22. Дан список произвольной структуры, каждый атом встречается по одному разу. Написать функцию, которая для каждого атома определяет его уровень вложенности и строит соответствующий список:
 $(A ((B) C)) \longrightarrow ((A 1) (B 3) (C 2))$
23. Найти сумму первых k числовых атомов списка произвольной структуры.
24. Построить список, определяющий сколько раз встречается каждый атом в списке: $(A (B A C B) C (A (B)) D) \longrightarrow ((A 3) (B 3) (C 2) (D 1))$
25. Дан список произвольной структуры. Найти наиболее часто встречающийся атом в списке.
26. Дано множество, представленное списком. Построить все перестановки этого множества:
 $(A B C) \longrightarrow ((A B C) (A C B) (B A C) (B C A) (C A B) (C B A))$
27. Дано множество, представленное списком, и число K . Построить все сочетания из данного множества длиной K .
 $(A B C D) 3 \longrightarrow ((A B C) (A B D) (B C D) (A C D))$
28. Для произвольного списка вычислить сумму чисел на заданной глубине
 $((1 A) ((3 4) (D C)) (2) (((7 8))))$, 2 \rightarrow 3

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ №4

ПОСТРОЕНИЕ ИТЕРАЦИОННЫХ ПРОГРАММ ДЛЯ ОБРАБОТКИ СПИСКОВ

Задание: Реализовать обработку списков с использованием функции LOOP для вариантов из лабораторной работы № 2.

Задания выполняются в среде cLisp (или другой). Код программ и тестовые примеры привести в отчете.

Быть готовым пояснить любой вопрос по тексту кода.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ №5

СОЗДАНИЕ СПИСКОВ С ИСПОЛЬЗОВАНИЕ СПИСКОВЫХ ВКЛЮЧЕНИЙ И ПРОСТЕЙШАЯ ОБРАБОТКА С ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ LAMBDA И MAP

Задание 1: Используя списковые включения (list comprehension) сгенерировать двумерный список (возможно в 2-3 этапа) порядка n (для некоторых заданий порядок должен быть нечетным) и выполнить его обработку с использованием функции `lambda`

Варианты:

1.

$$T = \begin{pmatrix} 3 & 2 & 0 & 2 & 3 \\ 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 \\ 3 & 2 & 0 & 2 & 3 \end{pmatrix}$$

Вернуть одномерный список из сумм средних элементов строк (для четного n)

2.

$$T = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 2 & 3 \\ 4 & 3 & 2 & 1 & 2 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

Вернуть одномерный список из сумм средних элементов столбцов

3.

$$T = \begin{pmatrix} 5 & 4 & 3 & 2 & 1 \\ 6 & 7 & 8 & 9 & 10 \\ 15 & 14 & 13 & 12 & 11 \\ & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \end{pmatrix}$$

Вернуть одномерный список из N -го, $(N-1)$ -го, ... первого элементов

строк

4.

$$T = \begin{pmatrix} 5 & 4 & 3 & 2 & 1 \\ 4 & 5 & 4 & 3 & 2 \\ 3 & 4 & 5 & 4 & 3 \\ 2 & 3 & 4 & 5 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

Вернуть одномерный список из площадей прямоугольников, основание и высота: первый и последний элемент столбцов сгенерированной матрицы

5.

$$T = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 10 & 9 & 8 & 7 & 6 \\ 11 & 12 & 13 & 14 & 15 \\ & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \end{pmatrix}$$

Вернуть одномерный список из произведений средних элементов строк (для четного n)

6.

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 9 \\ 0 & 0 & 3 & 8 & 10 \\ 0 & 4 & 7 & 11 & 14 \\ 5 & 6 & 12 & 13 & 15 \end{pmatrix}$$

Вернуть одномерный список из площадей треугольников, основание и высота: первый и второй элемент столбцов сгенерированной матрицы

7.

$$T = \begin{pmatrix} 1 & 1 & 3 & 5 & 9 \\ 0 & 2 & 2 & 5 & 8 \\ 0 & 0 & 3 & 3 & 7 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \end{pmatrix}$$

Вернуть одномерный список из сумм первого и последнего элементов столбцов

8.

$$T = \begin{pmatrix} 1 & 2 & 4 & 8 & 16 \\ 0 & 1 & 2 & 4 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 4 & 0 \\ 1 & 2 & 4 & 8 & 16 \end{pmatrix}$$

Вернуть одномерный список из первых, вторых, третьих ... элементов столбцов

9.

$$T = \begin{pmatrix} 1 & 3 & 6 & 10 & 15 \\ 2 & 5 & 9 & 14 & 19 \\ 4 & 8 & 13 & 18 & 22 \\ 7 & 12 & 17 & 21 & 24 \\ 11 & 16 & 20 & 23 & 25 \end{pmatrix}$$

Вернуть одномерный список из площадей прямоугольников, основание и высота которых это первый и последний элемент строк сгенерированной матрицы

10.

$$T = \begin{pmatrix} 1 & 3 & 5 & 7 & 9 \\ 3 & 1 & 3 & 5 & 7 \\ 5 & 3 & 1 & 3 & 5 \\ & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \end{pmatrix}$$

Вернуть одномерный список из площадей треугольников, основание и высота которых это первый и второй элемент строк сгенерированной матрицы

11.

$$T = \begin{pmatrix} 1 & 3 & 5 & 7 & 9 & 11 \\ 2 & 4 & 6 & 8 & 10 & 12 \\ 3 & 5 & 7 & 9 & 11 & 13 \\ 4 & 6 & 8 & 10 & 12 & 14 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Вернуть одномерный список из сумм средних элементов строк (для четного n)

12.

$$T = \begin{pmatrix} 1 & 0 & 2 & 0 & 3 & 0 & \cdot & \cdot \\ 0 & 1 & 0 & 2 & 0 & 3 & \cdot & \cdot \\ 2 & 0 & 1 & 0 & 2 & 0 & \cdot & \cdot \\ & & \cdot & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \cdot & & \end{pmatrix}$$

Вернуть одномерный список из произведений первого и последнего элементов строк (для нечетного n)

13.

$$T = \begin{pmatrix} 1 & 2 & 4 & 6 & \dots \\ 3 & 1 & 2 & 4 & \dots \\ 5 & 3 & 1 & 2 & \dots \\ 7 & 5 & 3 & 1 & \dots \\ \cdot & \cdot & \cdot & \cdot & \dots \end{pmatrix}$$

Вернуть одномерный список из диагональных элементов строк

14.

$$T = \begin{pmatrix} 1 & 3 & 5 & 7 & 9 \\ 4 & 1 & 3 & 5 & 7 \\ 5 & 4 & 1 & 3 & 5 \\ 9 & 5 & 4 & 1 & 3 \\ 14 & 9 & 5 & 4 & 1 \end{pmatrix}$$

Вернуть одномерный список из сумм первого и последнего элементов строк

15.

$$T = \begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 10 \end{pmatrix}$$

Вернуть одномерный список из первых, вторых, третьих ... элементов строк

Задание 2: Используя списковые включения (**list comprehension**) сгенерировать одномерный список. Выполнить обработку списка с использованием функции **map** и **lambda**.

Варианты:

1.
 - Сгенерировать список в котором чередуются случайные натуральные числа и произвольные сочетаний символов.
 - Каждый числовой атом списка необходимо увеличить на единицу.
2.
 - Сгенерировать список со случайными натуральными числами, кратными 3.
 - Сбросить в 0 нечетные числа
3.
 - Сгенерировать список в котором чередуются натуральные числа и случайные подстроки.
 - Каждый строковый элемент списка необходимо заменить на его длину.
4.
 - Сгенерировать список со случайными натуральными числами до 20.
 - Разделить пополам все четные элементы.
5.
 - Сгенерировать 2 списка одинаковой длины: со случайными числами от -5 до 5 и с повторяющейся последовательностью: 2,3,4
 - Определить функцию возведения элементов одного списка в степень

другого

6.
 - Сгенерировать 2 списка: состоящий из случайных строк и повторяющейся произвольной подстроки.
 - Склеить элементы этого списка по 2, используя элемент 2 списка как префикс.
7.
 - Получить список слов из заданной строки (использовать метод строк `split`)
 - Получить новый список, чередованием элементов списка и произвольной строки
8.
 - Сгенерировать двухуровневый список (матрицу)
 - Выпрямить список (сделать одноуровневым)
9.
 - Сгенерировать двухуровневый список (матрицу)
 - Получить новый список, состоящий из сумм подсписков
10.
 - Сгенерировать двухуровневый список (матрицу)
 - Получить новый список, состоящий из максимумов подсписков
11.
 - Сгенерировать список из натуральных чисел, а другой, состоящих из строк
 - Написать функцию, которая, чередуя элементы двух списков, строит новый список
12.
 - Сгенерировать список из подсписков длины k
 - Вычислить суммы элементов тех подсписков, которые начинаются с нечетного числа
13.
 - Сгенерировать список состоящий из чередующихся чисел 5 и случайных чисел до 1000
 - Получить новый список из трехзначных чисел исходного
14.
 - Сгенерировать 2 списка из случайных чисел

- Создать список из элементов вычитания 2-го списка из первого
15.
 - Сгенерировать список из случайных подстрок
 - Создать список из элементов, содержащих 'a'

Задание 3: Выполнить обработку списков через любые функции высших порядков: **map, reduce, filter, any** и т.п., а также **list comprehension**

Варианты:

1. Дан список из произвольных элементов. Найти наиболее часто встречающийся элемент.
2. Дублировать элементы списка количество дублей задается.
(a b c) => (a a a b b b c c c) .
3. Отсортировать список списков в соответствии с длиной подсписков:
((a b c) (d e) (f g h) (d e) (i j k l) (m n) (o)) =>
((o) (d e) (d e) (m n) (a b c) (f g h) (i j k l))
4. Отсортировать список списков в соответствии с длиной подсписков:
сортировка по частотности длины
((a b c) (d e) (f g h) (d e) (i j k l) (m n) (o)) =>
((i j k l) (o) (a b c) (f g h) (d e) (d e) (m n))
5. Проверить, является ли список палиндромом.
6. Определите функцию, которая в одноуровневом списке чисел, вычисляет количество отрицательных и количество положительных чисел.
(4 -8 6 -9 -7) -> (3 2) .
7. Вычислить количество чисел в произвольном списке
8. Выполнить попарные перестановки элементов списка
(a b d f b a c) -> (b a f d a b c)
9. Добавить заданный элемент s на каждое K-е место
10. Определить функцию, разбивающую список на пары:
(a b c d) -> ((a b) (c d))

11. Два множества представлены списками. Найти пересечение множеств.
12. Два множества представлены списками. Найти вычитание множеств.
13. Два множества представлены списками. Найти объединение множеств.
14. Из данного списка удалить все элементы, которые являются списками длиной более K.
15. Дан список, число и произвольное s-выражение. Поставить s-выражение в список перед элементом, номер которого равен числу:
 $(a\ b\ c\ d),\ t,\ 3 \rightarrow (a\ b\ t\ c\ d)$
16. Написать функцию, превращающую список в множество, т.е. удаляющую все повторяющиеся элементы:
 $(a\ b\ a\ b\ b\ c\ a) \rightarrow (a\ b\ c)$
17. Написать функцию, проверяющую, является ли список множеством, т.е. каждый элемент в нем должен встречаться по одному разу.
18. Проверить, состоят ли два списка из одних и тех же элементов.
19. Дан список и число N. Разбить список на подсписки длины N:
 $(a\ b\ c\ d\ 1\ 2)\ 4 \rightarrow ((a\ b)\ (c\ d)\ (1\ 2))$
20. Дан одноуровневый список, состоящий из чисел. Построить список из последовательных сумм элементов.
 $(a_1, a_2, a_3, \dots, a_n) \rightarrow (a_1, a_1+a_2, a_1+a_2+a_3, \dots, a_1+a_2+\dots+a_n)$
21. Для двухуровневого списка вернуть подсписок максимальной длины.
22. Для одноуровневого списка удалить элемент, стоящий следом за заданным
 $(a\ b\ a\ b\ b\ c\ a),\ a \rightarrow (a\ a\ b\ c\ a)$
23. Написать функцию, вычисляющую произведение всех элементов одноуровневого числового списка в предположении, что отсутствует операция умножения и можно использовать лишь сложение.
24. Из произвольного списка и числового списка построить новый список:
 $(a\ b\ c)\ (1\ 2\ 3) \rightarrow ((a)\ (b\ b)\ (c\ c\ c))$

25. Даны два множества, представленные списками. Проверить, является ли первое множество подмножеством второго.
26. Построить список, определяющий сколько раз встречается каждый элемент в списке:
 $(a\ b\ a\ c\ b\ c\ a\ b\ d) \rightarrow ((a\ 3)\ (b\ 3)\ (c\ 2)\ (d\ 1))$
27. Даны два числовых списка. Построить третий список, количество элементов которого равно длине второго списка и каждый элемент равен количеству элементов первого списка, меньших соответствующего элемента второго списка:
 $(1\ 7\ 3\ 0\ 2)\ (3\ 7\ 10) \rightarrow (3\ 4\ 5)$
28. Написать функцию, удаляющую из списка каждый K-й элемент.
29. Дан список и числа N и M. Вырезать из списка подсписок от элемента N до элемента M:
 $(a\ b\ c\ d\ e)\ 2\ 4 \rightarrow (b\ c\ d)$
30. Есть список целых чисел. Требуется получить из него список пар с началами и концами непрерывных подпоследовательностей.
 Например,
 $(1,\ 2,\ 3,\ 5,\ 6,\ 11) \rightarrow ((1,\ 3),\ (5,\ 6),\ (11,\ 11))$
31. Даны два списка чисел, которые могут содержать до 100000 чисел каждый. Посчитайте, сколько чисел содержится одновременно как в первом списке, так и во втором.
32. Даны два списка чисел, которые могут содержать до 10000 чисел каждый. Выведите все числа, которые входят как в первый, так и во второй список, в порядке возрастания.
33. Дан список чисел. Выведите все элементы списка, которые больше предыдущего элемента.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ №6

РЕКУРСИВНЫЕ ВЫЧИСЛЕНИЯ. МЕМОИЗАЦИЯ.

Задание: Построить рекурсивную функцию, выполнить замеры расхода памяти и времени. (datetime или timeit, sys.deep_getsizeof())

Сделать реализацию с мемоизацией. Получить время и память.

Сделать реализацию с декоратором @lru_cache (модуль functools).
Получить время и память.

1. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:

$$F(n) = n \text{ при } n \leq 3;$$

$$F(n) = n * n * n + F(n - 1), \text{ если } n > 3 \text{ и дает остаток } 0 \text{ при делении на } 3$$

$$F(n) = 4 + F(n // 3), \text{ если } n > 3 \text{ и дает остаток } 1 \text{ при делении на } 3$$

$$F(n) = n * n + F(n - 2), \text{ если } n > 3 \text{ и дает остаток } 2 \text{ при делении на } 3$$

2. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:

$$F(n) = 5 - n \text{ при } n < 5$$

$$F(n) = 4 \cdot (n - 5) \cdot F(n - 5), \text{ если } n \geq 5 \text{ и } n \text{ делится на } 3,$$

$$F(n) = 3n + 2 \cdot F(n - 1) + F(n - 2), \text{ если } n \geq 5 \text{ и } n \text{ не делится на } 3.$$

3. Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(n) = 1 \text{ при } n = 1;$$

$$F(n) = n + F(n - 1), \text{ если } n - \text{чётно},$$

$$F(n) = 2 \times F(n - 2), \text{ если } n > 1 \text{ и при этом } n - \text{нечётно}.$$

4. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:

$$F(n) = n + 3, \text{ при } n \leq 18$$

$$F(n) = (n // 3) \cdot F(n // 3) + n - 12, \text{ при } n > 18, \text{ кратных } 3$$

$$F(n) = F(n - 1) + n \cdot n + 5, \text{ при } n > 18, \text{ не кратных } 3$$

5. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:

$$F(n) = n \cdot n + 5 \cdot n + 4, \text{ при } n > 30$$

$$F(n) = F(n + 1) + 3 \cdot F(n + 4), \text{ при чётных } n \leq 30$$

$$F(n) = 2 \cdot F(n + 2) + F(n + 5), \text{ при нечётных } n \leq 30$$

6. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = 1$ при $n \leq 1$;
 $F(n) = n \cdot F(n - 1)$ при чётных $n > 1$;
 $F(n) = n + F(n - 2)$ при нечётных $n > 1$;
7. Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:
 $F(0) = 1, F(1) = 1$
 $F(n) = 3 \cdot F(n-1) - F(n-2)$, при $n > 1$
8. Алгоритм вычисления значения функции $F(n)$, где n – натуральное число,
задан следующими соотношениями:
 $F(0) = 1, F(1) = 3, F(2) = 2$
 $F(n) = F(n-1) * F(n-3)$, при $n > 2$
9. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = -n$ при $n < 0$
 $F(n) = 2n + 1 + F(n-3)$, если n чётно,
 $F(n) = 4n + 2 \cdot F(n-4)$, если n нечётно.
10. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = 5$ при $n = 0$,
 $F(n) = 3 \cdot F(n-4)$, если n положительное,
 $F(n) = F(n+3)$, если n отрицательное.
11. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = F(n+2) + 2 \cdot F(n*3)$, при $n \leq 70$
 $F(n) = n - 50$, при $n > 70$.
12. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = 1$, при $n < -100000$,
 $F(n) = F(n - 1) + 3 \cdot F(n - 3) + 2$, при $n > 10$,
 $F(n) = -F(n - 1)$ для остальных случаев.

13. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = n$ при $n \leq 3$;
 $F(n) = n // 4 + F(n-3)$ при $3 < n \leq 32$;
 $F(n) = 2 \cdot F(n-5)$ при $n > 32$
14. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = n$ при $n \leq 10$;
 $F(n) = n // 4 + F(n-10)$ при $10 < n \leq 36$;
 $F(n) = 2 \cdot F(n-5)$ при $n > 36$
15. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = 1$ при $n \leq 1$;
 $F(n) = n \cdot F(n-1)$ при чётных $n > 1$;
 $F(n) = n + F(n-2)$ при нечётных $n > 1$;
16. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = 1$ при $n \leq 1$;
 $F(n) = n + F(n-1)$ при чётных $n > 1$;
 $F(n) = n \cdot n + F(n-2)$ при нечётных $n > 1$;
17. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = 2$ при $n \leq 1$
 $F(n) = F(n-1) + F(n-2) + 2n + 4$, если $n > 1$
18. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = 1+2n$ при $n < 5$
 $F(n) = 2 \cdot (n+1) \cdot F(n-2)$, если $n \geq 5$ и n делится на 3,
 $F(n) = 2 \cdot n + 1 + F(n-1) + 2 \cdot F(n-2)$, если $n \geq 5$ и n не делится на 3.
19. Алгоритм вычисления функции $F(n)$ задан следующими соотношениями:
 $F(n) = -n$ при $n < 0$
 $F(n) = 2n + 1 + F(n-3)$, если n чётно,
 $F(n) = 4n + 2 \cdot F(n-4)$, если n нечётно.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ № 7

Решение задач общего вида с использованием парадигмы

функционального программирования

1.	Допустим, требуется зашифровать сообщения, состоящие из определенных символов. Например, исходный текст состоит из цифры от 0 до 9 (представленных в программе в виде символов) и других знаков и имеет длину в 15 символов. В шифровке на место цифр подставляются соответствующие им знаки, на место всех остальных символов дополнительный 11-й знак.
2.	Дан массив из двадцати элементов. Найти самую длинную последовательность из нулей и вывести на экран ее длину и номер ее начала в массиве.
3.	<p>Дана следующая информация (однако, вы можете проверить ее самостоятельно):</p> <p>16. 1 января 1900 года - понедельник.</p> <p>17. В апреле, июне, сентябре и ноябре 30 дней. В феврале 28 дней, в високосный год - 29. В остальных месяцах по 31 дню.</p> <p>18. Високосный год - любой год, делящийся нацело на 4, однако последний год века (XX00) является високосным в том и только том случае, если делится на 400.</p> <p>Сколько воскресений выпадает на первое число месяца в двадцатом веке (с 1 января 1901 года до 31 декабря 2000 года)?</p>
4.	Способ возведения в квадрат двузначные (и даже некоторые трехзначные) числа, оканчивающиеся на 5 заключается в следующем: для возведения в квадрат числа, оканчивающегося на 5 достаточно умножить число, полученное из исходного вычеркиванием последней пятерки на следующее по порядку число, затем остается лишь приписать «25» к получившемуся результату справа. Например, для того, чтобы возвести число 125 в квадрат достаточно 12 умножить на 13 и приписать 25, т.е. приписывая к числу $12 \cdot 13 = 156$ число 25, получаем результат 15625, т.е. $125^2 = 15625$. Напишите программу, возводящую число, оканчивающееся на 5, в квадрат.

5.	Напишите программу, которая бы по заданному и предложенному числам сообщала количество быков и коров. (По правилам игры «Быки и коровы»)
6.	Двоичное число, введенное по запросу с клавиатуры, в случае правильной записи преобразовать в шестнадцатеричное и десятичное число и результат вывести на экран.
7.	Написать программу определения количества шестизначных "счастливых" трамвайных билетов, у которых сумма первых трех цифр совпадает с суммой трех последних.
8.	На входе - текст. Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов. Определите, сколько различных слов содержится в этом тексте.
9.	На вход подаются число в двоичной системе счисления. Выполнить кодирование Грея.
10	Разложить багаж заданного объема по ячейкам заданных размеров (размер характеризуется объемом). Выведите объем багажа, который можно разместить в ячейках.
11	<p>XOR для произвольного числа аргументов определяется следующим образом:</p> $\text{xor}(a_1 a_2 \dots, a_n) = \text{xor}(a_1 \text{ xor}(a_2 \text{ xor}(a_3 \dots \text{ xor}(a_n)) \dots))$ <p>Посчитайте XOR от n последовательностей равной длины k.</p> <p>Формат ввода</p> <p>На первой строке записано число $2 \leq n \leq 1000$ — количество последовательностей.</p> <p>На последующих n строках записаны последовательности A_1, \dots, A_n из 0 и 1, разделённых пробелами равной длины $1 \leq k \leq 1000$.</p> <p>Формат вывода</p> <p>Выведите последовательность $C = \text{xor}(A_1, \dots, A_n)$, разделяя числа последовательности пробелами.</p>

12	<p>Выведите все простые на отрезке $[2;n]$.</p> <p>Формат ввода</p> <p>Вводится число $2 \leq n \leq 100000$.</p> <p>Формат вывода</p> <p>Выведите все простые числа из отрезка $[2,n]$ в порядке возрастания</p> <p>Примечания</p> <p>Напомним, что проверить число на то, простое ли оно можно за количество операций порядка \sqrt{N}. Также напомним, что функция <code>math.sqrt</code> работает значительно быстрее, чем $(x ** 1/2)$.</p>
13	<p>Перед началом тараканьих бегов всем болельщикам было предложено сделать по две ставки на результаты бегов. Каждая ставка имеет вид "Таракан №А придет раньше, чем таракан №В". Организаторы бегов решили выяснить, могут ли тараканы прийти в таком порядке, чтобы у каждого болельщика сыграла ровно одна ставка из двух (то есть чтобы ровно одно из двух утверждений каждого болельщика оказалось верным). Считается, что никакие два таракана не могут прийти к финишу одновременно.</p> <p>Формат ввода</p> <p>В первой строке входных данных содержатся два разделенных пробелом натуральных числа: число K, не превосходящее 10, - количество тараканов и число N, не превосходящее 100, - количество болельщиков. Все тараканы пронумерованы числами от 1 до K. Каждая из следующих N строк содержит 4 натуральных числа A, B, C, D, не превосходящих K, разделенных пробелами. Они соответствуют ставкам болельщика "Таракан №А придет раньше, чем таракан №В" и "Таракан №С придет раньше, чем таракан №D".</p> <p>Формат вывода</p> <p>Если завершить бега так, чтобы у каждого из болельщиков сыграла ровно одна из двух ставок, можно, то следует вывести номера тараканов в том порядке, в котором они окажутся в итоговой таблице результатов (сначала номер таракана, пришедшего первым, затем номер таракана, пришедшего вторым и т. д.) в одну строку через пробел. Если таких вариантов несколько, выведите любой из них. Если требуемого результата добиться нельзя, выведите одно число 0.</p>
14	<p>Во входной строке записана последовательность чисел через пробел.</p>

	<p>Для каждого числа выведите слово YES (в отдельной строке), если это число ранее встречалось в последовательности или NO, если не встречалось.</p>
15	<p>На Новом проспекте для разгрузки было решено пустить два новых автобусных маршрута на разных участках проспекта. Известны конечные остановки каждого из автобусов. Определите количество остановок, на которых можно пересест с одного автобуса на другой.</p> <p>Формат ввода</p> <p>Вводятся четыре числа, не превосходящие 100, задающие номера конечных остановок. Сначала для первого, потом второго автобуса (см. примеры и рисунок).</p> <p>Формат вывода</p> <p>Ваша программа должна выводить одно число — искомое количество остановок.</p> <pre>a, b, c, d = map(int, input().split()) print(len([item for item in range(min(a, b), max(a, b) + 1) if item in range(min(c, d), max(c, d) + 1)])) # ну, как бы все)</pre>
16	<p>В обувном магазине продается обувь разного размера. Известно, что одну пару обуви можно надеть на другую, если она хотя бы на три размера больше. В магазин пришел покупатель. Требуется определить, какое наибольшее количество пар обуви сможет предложить ему продавец так, чтобы он смог надеть их все одновременно.</p> <p>Формат ввода</p> <p>Сначала вводится размер ноги покупателя (обувь меньшего размера он надеть не сможет), в следующей строке — размеры каждой пары обуви в магазине через пробел. Размер — натуральное число, не превосходящее 100.</p> <p>Формат вывода</p> <p>Выведите единственное число — максимальное количество пар обуви, которое сможет надеть покупатель.</p>
17	<p>Системный администратор вспомнил, что давно не делал архива пользовательских файлов. Однако, объем диска, куда он может поместить</p>

	<p>архив, может быть меньше чем суммарный объем архивируемых файлов.</p> <p>Известно, какой объем занимают файлы каждого пользователя.</p> <p>Напишите программу, которая по заданной информации о пользователях и свободному объему на архивном диске определит максимальное число пользователей, чьи данные можно поместить в архив.</p> <p>Формат ввода</p> <p>Программа получает на вход в одной строке число S – размер свободного места на диске (натуральное, не превышает 10000), и число N – количество пользователей (натуральное, не превышает 100), после этого идет N чисел - объем данных каждого пользователя (натуральное, не превышает 1000), записанных каждое в отдельной строке.</p> <p>Формат вывода</p> <p>Выведите наибольшее количество пользователей, чьи данные могут быть помещены в архив.</p>
18	<p>Дан список. Определите, является ли он монотонно возрастающим (то есть верно ли, что каждый элемент этого списка больше предыдущего). Выведите YES, если массив монотонно возрастает и NO в противном случае. Решение оформите в виде функции <code>IsAscending(A)</code>. В данной функции должен быть один цикл <code>while</code>, не содержащий вложенных условий и циклов — используйте схему линейного поиска.</p>
19	<p>Дан список чисел. Определите, сколько в этом списке элементов, которые больше двух своих соседей и выведите количество таких элементов.</p> <p>Формат ввода</p> <p>Вводится список чисел. Все числа списка находятся на одной строке.</p>
20	<p>Дан список из чисел и индекс элемента в списке k. Удалите из списка элемент с индексом k, сдвинув влево все элементы, стоящие правее элемента с индексом k.</p> <p>Программа получает на вход список, затем число k. Программа сдвигает все элементы, а после этого удаляет последний элемент списка при помощи метода <code>pop()</code>.</p> <p>Программа должна осуществлять сдвиг непосредственно в списке, а не делать это при выводе элементов. Также нельзя использовать дополнительный список.</p> <p>Формат ввода</p>

	Вводится список чисел. Все числа списка находятся на одной строке. В следующей строке вводится одно целое число.
21	<p>Напишите программу, которая находит в массиве элемент, самый близкий по величине к данному числу.</p> <p>Формат ввода</p> <p>В первой строке задается одно натуральное число N, не превосходящее 1000 – размер массива. Во второй строке содержатся N чисел – элементы массива (целые числа, не превосходящие по модулю 1000). В третьей строке вводится одно целое число x, не превосходящее по модулю 1000.</p>

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ № 8

РЕКУРСИВНАЯ ОБРАБОТКА СПИСКОВ НА ПРОЛОГЕ

Задание: Написать базу, для рекурсивной обработки списков в среде SWI-Prolog.

Код программ и тестовые примеры привести в отчете. Быть готовым пояснить любой вопрос по тексту кода.

Варианты:

Варианты заданий взять те же, что и для лабораторной работы № 2.