

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**  
**Кафедра програмування**

**Затверджено**

На засіданні кафедри програмування  
факультету прикладної математики  
та інформатики  
Львівського національного університету  
імені Івана Франка  
(протокол № 1 від 29 серпня 2023 р.)



Зав. кафедри к. ф.-м. н., доц. Ярошко С. А.

**Силабус з навчальної дисципліни**  
**«Логічне та функціональне програмування»,**  
**що викладається в межах ОПІ Інформатика**  
**першого (бакалаврського) рівня вищої освіти для здобувачів**  
**зі спеціальності 122 Комп'ютерні науки**

Львів 2023 р.

<b>Назва дисципліни</b>	Логічне та функціональне програмування
<b>Адреса викладання дисципліни</b>	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики, кафедра програмування
<b>Галузь знань, шифр та назва спеціальності</b>	Галузь знань: 12 Інформаційні технології Спеціальність: 122 Комп'ютерні науки
<b>Викладачі дисципліни</b>	Ярошко Сергій Адамович, к. ф.-м. н., доцент, завідувач кафедри програмування
<b>Контактна інформація викладачів</b>	Електронна пошта: serhiy.yaroshko@lnu.edu.ua, веб-сторінка: <a href="https://ami.lnu.edu.ua/employee/yaroshko">https://ami.lnu.edu.ua/employee/yaroshko</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі он-лайн консультації через Microsoft Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
<b>Сторінка курсу</b>	<a href="https://ami.lnu.edu.ua/course/logical-and-functional-programming-informatics">https://ami.lnu.edu.ua/course/logical-and-functional-programming-informatics</a>
<b>Інформація про дисципліну</b>	Дисципліна «Логічне та функціональне програмування» є нормативною дисципліною зі спеціальності 122 Комп'ютерні науки (інформатика) для освітньої програми Комп'ютерні науки, яку викладають у сьомому семестрі в обсязі 3 кредити (за Європейською кредитно-трансферною системою ECTS)
<b>Коротка анотація дисципліни</b>	Фокус уваги курсу спрямовано на вивчення засад декларативного програмування. Засоби та методи написання програм у логічному стилі проілюстровано мовою Prolog, засоби та методи функціонального програмування – з використанням Lisp і Python. Для опису задач логічний стиль застосовує твердження і правила, функціональний – оголошення та виклики функцій. Розглянуто питання правильної побудови рекурсії, використання монад. Курс пов'язаний з дисциплінами "Програмування", "Теорія алгоритмів", "Алгоритми і структури даних".
<b>Мета та цілі дисципліни</b>	Метою вивчення нормативної дисципліни «Логічне та функціональне програмування» є набуття теоретичних знань і практичних умінь для застосування альтернативного (порівняно з імперативним) підходу до побудови програм.
<b>Література для вивчення дисципліни</b>	<p><i>Основна література</i></p> <ol style="list-style-type: none"> <li>1. SWI Prolog Reference manual – Електронний ресурс. Режим доступу: <a href="https://www.swi-prolog.org/pldoc/doc_for?object=manual">https://www.swi-prolog.org/pldoc/doc_for?object=manual</a></li> <li>2. Prolog Tutorial – Електронний ресурс. Режим доступу: <a href="https://www.tutorialspoint.com/prolog/index.htm">https://www.tutorialspoint.com/prolog/index.htm</a></li> <li>3. Ivan Bratko Prolog Programming For Artificial Intelligence. Fourth Edition – Addison Wesley, 2021. – 697 с.</li> <li>4. Шумейко О. О. Visual Prolog. Опануй на прикладах : навч. посіб. / О. О. Шумейко, В.М. Кнуренко. – Дніпропетровськ : Біла К. О., 2014. – 404 с.</li> <li>5. Learn Lisp – Електронний ресурс. Режим доступу: <a href="https://www.tutorialspoint.com/lisp/index.htm">https://www.tutorialspoint.com/lisp/index.htm</a></li> <li>6. Steven Lott Functional Python Programming – Packt Publishing Ltd., 2015 – 339 р.</li> </ol> <p><i>Додаткова література</i></p> <ol style="list-style-type: none"> <li>1. Learn Common Lisp – Електронний ресурс. Режим доступу: <a href="https://lisp-lang.org/learn/">https://lisp-lang.org/learn/</a></li> <li>2. Christian Neumanns Simple Introduction to Monads - CodeProject [Електронний ресурс] – Режим доступу: <a href="https://www.codeproject.com/Articles/5290753/Simple-Introduction-to-Monads">https://www.codeproject.com/Articles/5290753/Simple-Introduction-to-Monads</a></li> <li>3. David Mertz Functional Programming in Python – O'Reilly Media, Inc., 2015 – 39 р.</li> <li>4. Online courses – <a href="https://www.classcentral.com/report/best-free-prolog-courses/">https://www.classcentral.com/report/best-free-prolog-courses/</a></li> </ol>

<b>Обсяг курсу</b>	90 (90) годин аудиторних занять. З них 32 години лекцій, 32 години лабораторних занять та 26 годин самостійної роботи
<b>Очікувані результати навчання</b>	<p>Після завершення цього курсу студент буде :</p> <p><b>Знати</b></p> <ul style="list-style-type: none"> <li>- синтаксис мови Prolog, засади функціонування її машини виводу, вбудовані засоби;</li> <li>- синтаксис мови Lisp та особливості функціонування її інтерпретатора, стандартні функції;</li> <li>- прийоми написання програм у логічному стилі;</li> <li>- правила написання функціональних програм.</li> </ul> <p><b>Вміти</b></p> <ul style="list-style-type: none"> <li>- описувати предметну область задачі мовою Prolog;</li> <li>- формулювати прості та рекурсивні правила мовою Prolog;</li> <li>- створювати та використовувати структури (у тому числі й бази даних) мовою Prolog;</li> <li>- оголошувати та використовувати функції мовою Lisp, у тому числі функції вищих порядків;</li> <li>- застосовувати засоби функціонального програмування мови Python.</li> </ul>
<b>Компетентності</b>	<p><i>Інтегральна:</i> Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачають застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.</p> <p><i>Загальні (ЗК):</i></p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.  ЗК2. Здатність застосовувати знання у практичних ситуаціях.  ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.  ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.  ЗК8. Здатність генерувати нові ідеї (креативність).  ЗК9. Здатність працювати в команді.  ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.  СК2,3,4,6,7,8,10,11,16</p> <p><i>Спеціальні (фахові, предметні) компетентності (СК):</i></p> <p>СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.  СК7. Здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів.  СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.  СК17. Здатність розв'язувати прикладні задачі на основі інтегрованого знання про основні методи інформатики та інформаційні технології, використовувати комп'ютер і технології зв'язку, представляти дані у зрозумілій для всіх формі, яка проявляється у прагненні, здатності і готовності до ефективного застосування сучасних засобів інформаційних та комп'ютерних технологій для розв'язання завдань у професійній діяльності, усвідомлюючи при цьому значущість предмету і результату діяльності.</p>
<b>Програмні результати навчання</b>	<p>ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.  ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.</p>

	<p>PR9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.</p>				
<b>Ключові слова</b>	<p>Логічне програмування, функціональне програмування, Prolog, Lisp, твердження, правило, машина виведення, функція вищого порядку, лямбда-числення.</p>				
<b>Формат курсу</b>	<p>Очний.</p>				
<b>Теми</b>	Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)* *лекція, самостійна, дискусія, групова робота)	Завдання, год	Термін виконання
	1	Парадигми програмування: імперативне, декларативне. Логічне програмування в Prolog: факт, відношення, правило. Моделювання відношень реального світу.	Лекція	2	
		Prolog-програми для моделювання простих об'єктів реального світу. Середовище програмування. Трасування.	Лабораторне, групова робота	2	Наступне ЛЗ
		Встановлення середовища програмування Prolog / випробування онлайн-ресурсу	Самостійна робота	1	
	2	Декларативний і процедурний сенс програм, обмеження перебору. Структура даних "список". Процедури опрацювання списків.	Лекція	2	
		Моделювання генеалогічного дерева. Рекурсивні визначення правил. Порядок тверджень у правилах.	Лабораторне, групова робота	2	Наступне ЛЗ
		Експеримент зі зміною порядку тверджень у процедурі	Самостійна робота	2	
	3	Об'єкти даних Prolog: атоми, структури. Суміщення зразків. Машина виведення, повернення. Моделювання ігрових задач.	Лекція	2	
		Оголошення та використання функцій для опрацювання списків.	Лабораторне, групова робота	2	Наступне ЛЗ
		Порівняння звичайної та хвостової рекурсії	Самостійна робота	2	
	4	Відсікання та їхній вплив на сенс Prolog-програми. Заперечення як неуспіх. Розпізнавання типу терма. Підрахунок входжень (не суміщень). Числові ребуси.	Лекція	2	
		Моделювання поліномів-списків.	Лабораторне, групова робота	2	Наступне ЛЗ
		Порівняння червоних і зелених відсікань	Самостійна робота	1	
	5	Бази даних засобами Prolog. Отримання структурованої інформації з БД. Задача планування поїздки.	Лекція	2	
		Побудова та опрацювання баз даних мовою Prolog. Моделювання ієрархічної структури бази. Оголошення та використання сервісних функцій.	Лабораторне, групова робота	2	Наступне ЛЗ
		Моделювання глибоко структурованих даних. Побудова і використання селекторів.	Самостійна робота	1	
	6	Розв'язування задачі про розстановку восьми ферзів. Вплив обраних структур даних на ефективність програми. Надлишковість моделі як засіб підвищення ефективності.	Лекція	2	
		Побудова та опрацювання баз даних мовою Prolog. Алгоритми перебору всіх записів бази: рекурсивне накопичення всіх розв'язків, пошук екстремального.	Лабораторне, групова робота	2	Наступне ЛЗ
		Випробування різних способів перебирання всіх записів бази даних	Самостійна робота	2	
	7	Створення і декомпозиція термів. Підстановки. Конструювання цілей і запуск їхнього досягнення. "Правильна рекурсія".	Лекція	2	
		Prolog-програми для розв'язування логічних ігрових задач (про зважування, перевезення місіонерів і канібалів тощо). Пошук шляху в лабіринті. Задачі на графах.	Лабораторне, групова робота	2	Наступне ЛЗ
		Пошук цікавої логічної задачі	Самостійна робота	1	
8	Робота з базою знань Prolog-програми. Алгоритми отримання всіх можливих розв'язків. Загальні питання стилю написання Prolog-програм.	Лекція	2		
	Розв'язування популярних числових ребусів засобами Prolog. Шахові задачі.	Лабораторне, групова робота	2	Наступне ЛЗ	
	Моделювання шахових задач	Самостійна робота	1		
9		Функціональне програмування в Lisp: дані, примітивні функції, списки. Функції	Лекція	2	

		призначення. Рекурсивне та ітеративне визначення функцій.			
		Створення програм для розв'язування логічних задач підвищеної складності.	Лабораторне, групова робота	2	Наступне ЛЗ
		Створення презентації до розв'язку задачі.	Самостійна робота	2	
10		Атоми, конси, списки. Розпізнавачі типу. Функції властивостей. Числові та логічні функції. Контрольні конструкції Lisp.	Лекція	2	
		Оголошення символу "Me", наділення його властивостями. Оголошення простих функцій для доступу до структурованих властивостей.	Лабораторне, групова робота	2	Наступне ЛЗ
		Встановлення середовища програмування Lisp / випробування онлайн-ресурсу	Самостійна робота	1	
11		Засоби виведення. Задачі породження комбінаторних об'єктів. Лямбда-числення та функції вищих порядків.	Лекція	2	
		Оголошення рекурсивних функцій для роботи зі списками: лінійними та ієрархічними.	Лабораторне, групова робота	2	Наступне ЛЗ
		Експерименти з рекурсивними й ітеративними аналогами функцій	Самостійна робота	2	
12		Побудова бінарних дерев. Функції модифікації (зв'язних структур). Функції планування та їх застосування до груп послідовностей даних.	Лекція	2	
		Оголошення ітеративних функцій для роботи зі списками: лінійними та ієрархічними.	Лабораторне, групова робота	2	Наступне ЛЗ
		Експерименти з рекурсивними й ітеративними аналогами функцій	Самостійна робота	2	
13		Замикання засобами Lisp. Фабрики функцій. Карування функцій. Приклади використання.	Лекція	2	
		Функції та структури для моделювання розріджених поліномів, нечітких множин, багатовимірних масивів.	Лабораторне, групова робота	2	Наступне ЛЗ
		Моделювання деревовидних структур даних	Самостійна робота	2	
14		Оголошення структур мовою Lisp. Супутні функції. Наслідування структур. Функції опрацювання структур. ООП у Lisp.	Лекція	2	
		Функції вищих порядків: числове інтегрування, розв'язування алгебричних рівнянь, табулювання функцій.	Лабораторне, групова робота	2	Наступне ЛЗ
		Оголошення та використання функцій вищих порядків у Lisp.	Самостійна робота	2	
15		Засоби функціонального програмування мови Python: лямбда-вирази, відображення функцій на послідовності. Функції map, filter, reduce, apply.	Лекція	2	
		Використання функцій планування мови Lisp для отримання послідовності значень, для акумулювання результатів тощо. Порівняння з можливостями Python.	Лабораторне, групова робота	2	Наступне ЛЗ
		Порівняння вбудованих функцій вищих порядків Python і Lisp.	Самостійна робота	2	
16		Монади у функціональному програмуванні. Приклади реалізації та використання мовою Python. Бібліотека ruptionad.	Лекція	2	
		Приклади використання функціональних засобів мови Python.	Лабораторне, групова робота	2	На занятті
		Підготовка до іспиту	Самостійна робота	2	
<b>Підсумковий контроль, форма</b>	іспит в кінці семестру				
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з дисциплін "Програмування", "Алгоритми і структури даних", "Теорія алгоритмів" та вмінь знаходити необхідні документи в мережі інтернет, інсталювати та налаштовувати вільно поширювані середовища програмування.				
<b>Навчальні методи та техніки, які будуть використані під час викладання курсу</b>	Лекції, виклад парадигми декларативного програмування, розбір методів написання програм мовою Prolog, мовою Lisp. Лабораторні заняття у вигляді виконання практичних завдань і презентації отриманих результатів, обговорення написаних програм; самостійна робота з вивченням оприлюднених електронних матеріалів.				
<b>Необхідне обладнання</b>	Для проведення лекцій: комп'ютер, проектор, доступ до мережі інтернет. Для проведення лабораторних та виконання завдань: комп'ютер, ОС Windows/Linux,				

	доступ до інтернету, безкоштовні середовища програмування Strawberry Prolog Light, LispWorks, Python.
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<p>Студенти виконують не менше 10 індивідуальних завдань, презентують написані програми, пояснюють, які підходи та засоби при цьому використано. Виявлення ознак академічної недоброчесності в лабораторній роботі студента є підставою для її незарахування, незалежно від масштабів плагіату чи обману. Оцінювання проводиться за 100-бальною шкалою. 50 балів нараховують за виконання лабораторних завдань: по 5 балів за кожну з 10 лабораторних робіт. Ще 50 балів – за оволодіння теоретичним матеріалом курсу, з них 25 балів за презентацію лабораторних робіт і висвітлення важливих теоретичних моментів, а інші 25 – за підсумкове тестування.</p> <p>Активна участь у обговоренні лекційного матеріалу, самостійне відшукування та використання додаткової інформації для виконання робіт є підставою для нарахування додаткових балів.</p> <p><b>Відвідання занять</b> є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти мають інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів письмових робіт, передбачених курсом. Виконані роботи завантажують у відповідне хмарне сховище. Альтернативою відвідування лабораторних занять в університеті може бути дистанційна онлайн робота за розкладом проведення занять. Активність на лекціях і лабораторних урахують при оцінюванні відповідного лабораторного завдання.</p> <p>Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p> <p><b>Академічна доброчесність:</b> очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп'ютерних програм як своїх становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.</p>
<b>Опитування</b>	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.
<b>Запитання до іспиту</b>	<ol style="list-style-type: none"> <li>1. Особливості мови Prolog. Структура програми.</li> <li>2. Відношення, факт, правило, цільовий запит у мові Prolog.</li> <li>3. Як виглядає процедура Prolog-програми? Наведіть приклад.</li> <li>4. Оголошення простих правил, логічні діаграми відношень. Навести приклади.</li> <li>5. Рекурсивне визначення правил у мові Prolog, композиція відношень.</li> <li>6. Порядок побудови Prolog-системою відповіді на запит. Навести приклади.</li> <li>7. Декларативний і процедурний сенс Prolog-програм.</li> <li>8. Об'єкти даних мови Prolog.</li> <li>9. Суміщення Prolog-тверджень: визначення, правила виконання, приклади.</li> <li>10. Структура даних список мови Prolog: способи зображення списків, випадки використання, звертання до частин списку.</li> <li>11. Конкатенація списків мовою Prolog.</li> <li>12. Поділ списків на частини мовою Prolog.</li> <li>13. Обертання списку мовою Prolog.</li> <li>14. Визначення довжини списку мовою Prolog.</li> <li>15. Перевірка входження елемента до списку мовою Prolog.</li> <li>16. Вставка елемента до списку мовою Prolog.</li> <li>17. Вилучення елемента зі списку мовою Prolog.</li> <li>18. Відшукування найбільшого елемента списку мовою Prolog.</li> <li>19. Як визначити, чи містить список Prolog парну кількість елементів без обчислення.</li> <li>20. Впорядкування списку Prolog методом простої вставки.</li> <li>21. Чим відрізняється використання відношень мови Prolog від викликів процедур імперативних мов програмування?</li> <li>22. Порядок виконання арифметичних обчислень у програмах мовою Prolog.</li> </ol>

23. Подання структурованої інформації у Prolog-програмі.
24. Отримання інформації з бази даних мовою Prolog: утиліти доступу, селектори. Приклади запитів до бази.
25. Перебір усіх записів бази даних: відшукування екстремального значення. Приклад.
26. Перебір усіх записів бази даних: накопичення результатів за допомогою власних процедур. Приклад.
27. Перебір усіх записів бази даних: накопичення результатів за допомогою вбудованих функцій. Приклад.
28. Керування перебором: обмеження перебору, успіх, невдача (неуспіх).
29. Труднощі, пов'язані з використанням відсікань і заперечень у Prolog-програмах.
30. Як визначити мовою Prolog твердження "Мері любить усіх тварин крім змій"?
31. Перевірка типу термів мовою Prolog, перевірка входження елемента у список (не суміщення).
32. Створення і декомпозиція термів мовою Prolog.
33. Відношення підстановки для формул мовою Prolog.
34. Конструювання мети, досягнення сконструйованої мети мовою Prolog.
35. Маніпуляції з базою знань програми мовою Prolog.
36. Як мовою Prolog власноруч побудувати предикат *findAll*?
37. Розв'язування задачі «про мавпу і банан» мовою Prolog [3], накопичення ходів.
38. Розв'язування задачі про ханойські вежі мовою Prolog.
39. Розв'язання числових ребусів [3].
40. Задача планування поїздки мовою Prolog [3].
41. Розв'язання задачі про вісім ферзів (пошук списку пар  $x/y$ ) [3].
42. Розв'язання задачі про вісім ферзів (пошук перестановки ординат) [3].
43. Розв'язання задачі про вісім ферзів (подання з чотирма координатами) [3].
44. Оголошіть мовою Prolog відношення «масштабування» для двох списків термів: усім числовим аргументам термів першого списку відповідають аргументи, помножені на заданий коефіцієнт у термах другого списку; нечислові аргументи в обох списках збігаються.
45. Особливості, переваги функціонального програмування.
46. Структура Lisp-програми. Об'єкти мови Lisp. Інтерпретація об'єктів.
47. Списки, конси мови Lisp. Конструктори, крапкова нотація.
48. Примітивні функції Lisp. Комбінації CAR і CDR. Пояснити на прикладах.
49. Функції призначення мови Lisp.
50. Визначення власних функцій мовою Lisp. Прості та умовні завдання.
51. Як виконується тіло функції, у якому визначено декілька умовних і просте завдання? Поясніть на прикладі.
52. Оголошення мовою Lisp функції, яка перевіряє, чи належить символ списку.
53. Оголошення мовою Lisp функції, яка вилучає зі списку перше входження символу.
54. Оголошення мовою Lisp функції, яка вилучає зі списку всі входження символу.
55. Оголошення мовою Lisp функції, яка перевіряє рівність двох списків.
56. Оголошення мовою Lisp функції, яка виконує конкатенацію списків.
57. Оголошення мовою Lisp функції, яка обертає список.
58. Функції розпізнавання мови Lisp.
59. Функції мови Lisp для керування режимами виведення на консоль.
60. Арифметичні функції мови Lisp.
61. Алгебричні функції мови Lisp.
62. Логічні функції мови Lisp.
63. Функції порівняння мови Lisp.
64. Функції мови Lisp для роботи зі списками.
65. Функції керування ходом обчислень у Lisp – галуження.
66. Функції керування ходом обчислень у Lisp – повторення.
67. Функції керування ходом обчислень у Lisp – послідовність форм.
68. Властивості символів у Lisp: додавання/вилучення властивостей, читання/запис їхніх значень.
69. Як у Lisp реалізовано лямбда-числення, функції вищих порядків?
70. Функції виклику функцій у мові Lisp. Навести приклади використання.
71. Реалізація двійкових дерев засобами Lisp.
72. Функції обходу двійкового дерева мовою Lisp.
73. Функція додавання елемента до двійкового дерева мовою Lisp.
74. Функції модифікатора мови Lisp.
75. Функції планування MAPxxx мови Lisp. Навести приклади обчислень.
76. Функції фільтрування та агрегації мови Lisp.
77. Закриття мовою Lisp. Приклади використання.
78. Оголошення генераторів мовою Lisp.

79. Оголошення структур мовою Lisp. Конструктори, доступ до членів структури.
80. Наслідування структур мовою Lisp. Оголошення типу членів структури.
81. Оголошення класів мовою Lisp. Визначення методів.
82. Функціональні можливості мови Пайтон.  
приклади завдань на Lisp (не вичерпний перелік)
83. Побудуйте за допомогою вбудованих примітивів мови Lisp список, який містить два підсписки, перший з яких має три атоми, а другий – чотири атоми.
84. Побудуйте за допомогою вбудованих примітивів мови Lisp список, який містить три складені об'єкти, і лише його другий елемент є атомом.
85. Побудуйте за допомогою вбудованих примітивів мови Lisp список, у якого голова та хвіст є списками з підсписками
86. За допомогою вбудованих примітивів мови Lisp перетворити вхідні дані на результат: задано (A, B, C), (X, Y, Z); побудувати (A, Y, Z).
87. За допомогою вбудованих примітивів мови Lisp перетворити вхідні дані на результат:  
задано ((one 1) (two 2 3) (three 4 5 6)); побудувати 5.
88. Не використовуючи додаткових символів перетворити вхідні дані на результат:  
задано one=1, two=2, three=3; зробити: one=2, two=3, three=1.
89. Оголосіть функцію мовою Lisp, що розпізнає ієрархічні списки.
90. Оголосіть функцію мовою Lisp, що визначає глибину ієрархічного списку.
91. Оголосіть функцію мовою Lisp, що обчислює загальну кількість елементів ієрархічного списку.
92. Оголосіть функцію мовою Lisp, що перетворює список на множину.
93. Оголосіть функцію мовою Lisp, яка за вхідним списком будує вказану структуру:  
'(q w e r t y) → (q (w (e (r (t (y))))))
94. Оголосіть функцію мовою Lisp, яка за вхідним списком будує вказану структуру:  
'(q w e r t y) → ((((((y) t) r) e) w) q)
95. Оголосіть функцію мовою Lisp, яка за вхідним списком будує вказану структуру:  
'(q w e r t y) → ((q) (w) (e) (r) (t) (y))
96. Оголосіть функцію мовою Lisp, яка за вхідним списком будує вказану структуру:  
'(q w e r t y) → ((q w) (e r) (t y))
97. Оголосіть функцію мовою Lisp, яка перевіряє, чи впорядковано список чисел за зростанням (запропонуйте рекурсивний, ітеративний та послідовний варіанти).
98. Оголосіть функцію мовою Lisp, яка знаходить у списку перший елемент, який задовольняє заданий критерій (запропонуйте рекурсивний та ітеративний варіанти).
99. Оголосіть функцію мовою Lisp, яка застосовує задану функцію до кожного елемента списку (запропонуйте рекурсивний, ітеративний та послідовний варіанти).
100. Оголосіть функцію мовою Lisp, яка повертає частку та остачу від ділення x на y. Повернути результат у вигляді конса. Не використовувати функцій ділення та остачі.
101. Оголосіть функцію мовою Lisp, яка обчислює x в степені y. Запропонувати алгоритми з часовою оцінкою  $O(y)$  та  $O(\log_2 y)$ .
102. Оголосіть функцію мовою Lisp, яка до списку чисел застосовує список арифметичних функцій і обчислює значення утвореного виразу. Наприклад, '(3 6 4 1) '(+ \* -) → (3+6)\*4-1 = 35
103. Оголосіть функцію мовою Lisp, яка обчислює суму двох матриць за допомогою функцій планування.
104. Оголосіть функцію мовою Lisp, яка обчислює скалярний добуток двох векторів за допомогою функцій планування.