**Теми індивідуальних творчих робіт**

**з дисципліни «Основи програмування» (мова Python) для студентів 1-го курсу спеціальності 121 «інженерія програмного забезпечення»**

Індивідуальні творчі роботи не є обов’язковими. Вони призначені для демонстрації творчих можливостей студента та підвищення його рейтингу з програмування мовою Python.

Складність тем вказана за системою ECTS (A,B,C,D,E), де А – найвища складність, Е – найнижча складність. Максимальна кількість додаткових балів для завдань складності А дорівнює 20, для завдань складності D - 10.

Завдання не вимагає розробки графічного інтерфейсу користувача, веб-дизайну, вікон.

Для виконання тем з чисельних методів, матричної та векторної алгебри використання модуля **Numpy** заборонено. Для виконання завдань з обробки природної мови рекомендується використовувати модуль **NLTK.**

| **№ п.п** | **Назва теми** | **Складність** | **Прізвище студента-виконавця** |
| --- | --- | --- | --- |
|  | \*Мозковий штурм.  Програма генерує ідеї на основі асоціацій з описаною ситуацією, контрольних запитань, аналогій, .  https://tvoiekino.ru › blog › gde-raskapyvat-idei  <https://4brain.ru/blog/%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B-%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D0%B8%D0%B4%D0%B5%D0%B9/> | A |  |
|  | **Прогнозування результатів модульного та сесійного контролю знань студентів з дисципліни «Основи програмування».**  Програма здійснює прогноз екзаменаційної оцінки на основі поточних результатів успішності з дисципліни «Основи програмування» та споріднених з нею (Вища математика, Дискретна математика), використовуючи методи кластерного аналізу, <https://dspace.khadi.kharkov.ua/dspace/bitstream/123456789/1124/1/V_68_03.pdf> , машинного навчання <http://nz.uad.lviv.ua/static/media/1-58/10.pdf>, експертних оцінок <https://knowledge.allbest.ru/management/2c0b65635b2bd78a5c53b88421316c37_0.html> | A |  |
|  | **Автоматична класифікація документів (ділового листування)**  Програма аналізує повідомлення, класифікує їх за вказаною користувачем ознакою (лист-запит, лист-прохання, інформаційний лист, лист-відповідь, лист-підтвердження, рекомендаційний лист, лист-запрошення, лист-вітання, лист-подяка, лист-вибачення, лист-відмова, лист-нагадування, рекламний тощо), маркує їх певною міткою, розміщує їх по різних папках на дисках, здійснює швидкий пошук, відображає текст чи анотацію листа на екрані. - | A |  |
|  | **Автоматизація визначення переліку дисциплін за спеціальністю «Інженерія програмного забезпечення» на основі мотиваційного листа студента.**  Програма виділяє речення з текстового файлу, з речення виділяє слова, визначає та видаляє стоп-слова, визначає частини мови, розпізнає та будує залежності між словами в реченні, будує дерева залежностей слів, розпізнає іменовані сутності речення, видобуває факти з текстового файлу, будує потрібний контекст. <https://proglib.io/p/fun-nlp> | A |  |
|  | **Прогнозування вартості навчання.**  Програма прогнозує вартість навчання в різних університетах на різних спеціальностях в залежності від рівня інфляції, обсягу валового національного прибутку, середньої заробітної плати та інших показників. | A |  |
|  | \* Автоматизація створення переліку знань і умінь junior розробника ПЗ на онове вимог роботодавців. | AA |  |
|  | \* Create ICT Profiles based on European e-Competence Framework | AA |  |
|  | **Генератор паролів, логінів, ніків, афоризмів** тощо. <https://genword.ru/generators/password/> | B |  |
|  | \*Генерація афоризмів.  Програма генерує тексти на основі набору слів, які зберігаються у файлах відповідно до ІТ-тематиці (програмне забезпечення, мови програмування, комп’ютерні науки, алгоритми тощо). | B |  |
|  | **Програма перекладу коду з мови Python на мову С.** Програма зчитує код мовою Python з файлу, замінює мовні конструкції Python на мовні конструкції С, зберігає новий код у файлі і виводить змінений текст на екран | B |  |
|  | **Програма перекладу коду з мови С на мову Python.** Програма зчитує код мовою С з файлу, замінює мовні конструкції С на мовні конструкції Python, зберігає новий код у файлі і виводить змінений текст на екран. | B |  |
|  | **Навчальний тренажер з мови Python.**  Програма реалізує навігацію по Інтернет ресурсах, які містять інформацію про мову Python. Також програма виводить на екран завдання, приймає їх розв’язки від користувачів, визначає правильність відповідей, повідомляє про помилки, на вимогу користувача показує правильну відповідь, підсумовує результат тренування. | B |  |
|  | **«Шпаргалка з мови Python»**  Програма вибирає інформацію з мови Python з файлів. Класифікує її за важливістю. В діалозі з користувачем формує текст, таблиці, приклади для стислого подання теоретичного матеріалу та прикладів. | B |  |
|  | **Коллаборативна фільтрація з використанням косінусної міри подібності**.  Програма забезпечує пошук та підбір товарів на основі вподобань користувачів. Для цього користувач оцінює декілька подібних товарів. Робота рекомендаційної системи являє собою використання алгоритму фільтрації існуючих вподобань користувачів, для рекомендування нових наборів альтернатив, які є найбільш релевантними для конкретного користувача.  <https://habr.com/ru/post/150399/> | B |  |
|  | **Документатор коду.**  Програма генерує коментарі по коду програми і вставляє їх в потрібне місце коду. | C |  |
|  | **Навчальний тренажер по булевих функцій.**  Програма виводить на екран завдання, приймає їх розв’язки від користуачів, визначає правильність відповідей, повідомляє про помилки, на вимогу користувача показує правильну відповідь, підсумовує результат тренування. | C |  |
|  | **Електронний облік успішності студентів з дисципліни «Основи програмування».**  Програма накопичує показники (бали) поточної успішності студентів по лабораторних, самостійних, курсових розрахункових роботах в семестрі. Програма підсумовує результати та формує підсумкову відомість, яка зберігається у файлі. | C |  |
|  | **Пошук в текстовому файлі з використанням регулярних виразів.** <https://python-scripts.com/import-re-regular-expression> | C |  |
|  | **Обернені матриці та власні значення.**  Реалізація алгоритмів знаходження власного значення матриці та власного вектора, перетворення Хаусхолдера (QR -разложение), розрахунок оберненої матриці методом алгебраїчних доповнень та методом Гаусса. Порівняльний аналіз ефективності методів. | C |  |
|  | **Обчислення математичних виразів**  Реалізація алгоритму зворотного польського запису з перетворенням інфіксного виразу у постфіксний вираз (зворотній польський запис - ЗПЗ), обчислення постфіксного виразу (виразу по його ЗПЗ) та оптимізація виразів. <https://habr.com/ru/post/273253/> | C |  |
|  | **Апроксимація, інтерполяція, екстраполяція функцій**  Програма має реалізувати такі алгоритми:   * інтерполяційні сплайни; * інтерполяційні поліноми Лагранжа * інтерполяційні поліноми Ньютона | C |  |
|  | **Журнал куратора студентської групи.**  Програма повинна зберігати номери телефонів, ПІБ і адресу, студента і батьків, план виховних і культурно-масових заходів, облік їх виконання. Повинні бути передбачені введення нових даних і редагування наявних, пошук по ПІБ, за номером. При закритті програми і її перезапуску інформація повинна зберігатися в файлі. | C |  |
|  | **Генератор і розшифровщик абревіатур.**  Створити додаток, який генерує абревіатури та розшифровує їх. Для цього створити словник, ключем якого буде буква алфавіту, а значення список слів, що починаються на цю букву. При запуску програми користувачеві пропонується ввести будь-яку абревіатуру (набір символів), а на виході буде расшифровка. Наприклад, MASTIS - Establishing Modern Master-level Studies in Information Systems. Зворотна задача: ввести назву проекту, компанії тощо, на виході отримати абревіатуру. | C |  |
|  | **Щоденник студента.**  Список запланованих заходів - дата, час, тривалість, місце проведення. Автоматичне нагадування найближчої справи: за потокової дати та часом: відалення вчорашніх справ або перенесення на майбутнє. Аналіз "накладок" - перетинань запланованих справ. Перегляд справ на завтра, післязавтра тощо. | C |  |
|  | **Нечіткий пошук в тексті.**  Реалізація алгоритмів: нечіткого пошуку з індексацією, (Баеса-Ятеса — Гоннета), лінійний пошук з метрикою Левенштейна. Здійснити порівняльний аналіз ефективності алгоритмів. | C |  |
|  | **Бібліотека функцій математичної логіки**  Бібліотека складається з логічних функцій, що будують ДДНФ та ДКНФ булевої функції, що задана вектором, реалізують метод Блейка побудови скороченої ДНФ, метод Квайна-Мак-Класкі побудови скороченої ДНФ, будують імплікантну таблицю скороченої ДНФ, мінімальну ДНФ за імплікантною таблицею. | C |  |
|  | **Бібліотека функцій обробки множин та комбінаторики.**  Бібліотека складається з функцій, що оперують множинами (операції над скінченими множинами, перевірка рівності скінчених множин, перевірка рівності множин за допомогою таблиць належності, комбінаторні обчислення для основних теоретико-множинних операцій, побудова всіх перестановок заданої кількості елементів, побудова всіх розміщень заданої кількості елементів з заданої множини елементів, побудова всіх комбінацій заданої кількості елементів з заданої множини елементів) | C |  |
|  | **Створення вікторини.**  Розробити програму, що рандомно бере різні питання з файлу і з'єднує їх разом, створюючи вікторину для студентів. Потім програма має порівнювати відповіді з правильними варіантами. | D |  |
|  | **Планувальник подій і календар.**  Розробити програму, що дозволить користувачеві заносити дату і час події, а також деякі замітки в календар. Сам календар має зберігатися у файлі. Користувач повинен мати можливість переглянути календар або знайти певну подію. Програма може нагадувати користувачу про події відповідно до часу її настання. | D |  |
|  | **Система лінійних рівнянь.**  Написати програму для розв’язання систем лінійних рівнянь методом Гаусса. Вихідні дані і результати розв’язку зберігати у файлі. | D |  |
|  | **Система лінійних рівнянь.**  Написати програму для розв’язання систем лінійних рівнянь методом Зейделя. Вихідні дані і результати розв’язку зберігати у файлі. | D |  |
|  | **Система лінійних рівнянь.**  Написати програму розв’язання системи лінійних рівнянь методом Крамера. Кількість змінних до 20. Програма повинна виводити результати вирішення завдання, а також час, витрачений на розрахунок з точністю до сотих часток секунд. | D |  |
|  | **Визначник матриці**  Програма повинна створити (генерувати, ввести) матриці, обчислити визначник матриці, застосувавши алгоритм розкладанням матриці по рядках або стовпчиках, алгоритм Гауса приведення матриці до трикутного виду. | D |  |
|  | **Елементи векторної алгебри.**  Програма реалізує такі алгоритми:   * обчислення векторного та скалярного добутку * перевірка лінійної незалежності векторів * визначення паралельності та перпендикулярності векторів * множення вектора на число | D |  |
|  | **Матрична алгебра.**  Програма реалізує такі алгоритми:   * Множення довільної кількості матриць * Еквівалентні перетворення матриці (переставлення рядків, стовпчиків, множення рядків, стовпчиків на константу) | D |  |
|  | **Алгоритми множення матриць**  Реалізація алгоритмів:   * Алгоритм Штрассена - швидке множення матриць * Алгоритм Копперсмита – Винограду | D |  |
|  | **Чисельні методи розв’язання алгебричних рівнянь.**  Програма реалізує такі алгоритми: метод дотичних, метод хорд, метод простих ітерацій. Програма порівнює результати розв’язання алгебраїчних рівнянь і визначає ефективність алгоритмів. | D |  |
|  | **Система нелінійних рівнянь.**  Написати програму для розв’язання нелінійних рівнянь методом половинного ділення та методом Ньютона. Порівняти методи по швидкодії. | D |  |
|  | **Обчислення інтегралів.**  Написати програму обчислення інтегралів методами трапецій прямокутників і Сімпсона. Порівняти час реалізації методів при заданій точності. | D |  |
|  | **Пропозиція студента** | За погодженням з викладачем |  |

**Задачі складності D**

| **Задачі** | **Змістовна постановка задачі** | **Алгоритм розв’язання** | **Посилання** | **Прізвище студента-виконавця** |
| --- | --- | --- | --- | --- |
| 41. Задача про 8 ферзів | Задано ціле число **N**>0, що визначає кількість ферзів на шахівниці розміром **N**×**N**. Знайти усі можливі варіанти розміщення **N** ферзів таким чином, щоб вони не “погрожували” один одному. | Алгоритм перебору з поверненням ()  Задача має 92 розв’язків | <https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BF%D1%80%D0%BE_%D0%B2%D1%96%D1%81%D1%96%D0%BC_%D1%84%D0%B5%D1%80%D0%B7%D1%96%D0%B2>  Окулов С. М. Программирование в алгоритмах. — М.: БИНОМ. 2002 |  |
| 42. Обхід конем шахової дошки | Знайти маршруту [шахового коня](https://uk.wikipedia.org/wiki/%D0%9A%D1%96%D0%BD%D1%8C_(%D1%88%D0%B0%D1%85%D0%B8)), що проходить через усі поля [шахівниці](https://uk.wikipedia.org/wiki/%D0%A8%D0%B0%D1%85%D1%96%D0%B2%D0%BD%D0%B8%D1%86%D1%8F) по одному разу. | Алгоритм Ейлера, Алгоритм Вандермонда, | <https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BF%D1%80%D0%BE_%D1%85%D1%96%D0%B4_%D0%BA%D0%BE%D0%BD%D1%8F> |  |
| 43. Задача про лабіринт | Лабіринт являє собою прямокутник, що складається з одиничних клітинок. Деякі клітинки зайняті стіною, інші вільні. На одній з вільних клітинок знаходиться пацюк, на іншій - сало. Пацюк може пересуватись на сусідню клітинку (по горизонталі або вертикалі), але не може заходити у зайняті клітинки та виходити за межі лабіринту. Допоможіть пацюку знайти найкоротший шлях до сала. Якщо найкоротших шляхів декілька, необхідно знайти один будь-який. | Хвильовий алгоритм. | <https://disted.edu.vn.ua/courses/learn/310>  <https://uk.wikipedia.org/wiki/%D0%A5%D0%B2%D0%B8%D0%BB%D1%8C%D0%BE%D0%B2%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC> |  |
| 44. Задача про рюкзак | Для заданої множини N предметів, кожен з яких має вагу Wі та цінність Сi, визначити, яку кількість кожного з предметів слід взяти, так, щоб сумарна вага не перевищувала задану V, а сумарна цінність була максимальною. | Повний перебір  Метод гілок і меж  Метод динамічного програмування, Жадібний алгоритм, Генетичний алгоритм | <https://habr.com/ru/post/222577/>  <https://ru.wikibooks.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%BE%D0%B2/%D0%9A%D0%BE%D0%BC%D0%B1%D0%B8%D0%BD%D0%B0%D1%82%D0%BE%D1%80%D0%B8%D0%BA%D0%B0/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE_%D1%80%D0%B0%D0%BD%D1%86%D0%B5> |  |
| 45. Розбиття опуклого многокутника на трикутники | Задіано натуральне число N і опуклий многокутник з N+2 вершинами. Скільки існує способів розбити цей многокутник діагоналями, що не перетинаються, на трикутники? | Алгоритм розв’язання комбінаторних задач з використанням чисел Каталана | <https://neerc.ifmo.ru/wiki/index.php?title=%D0%A7%D0%B8%D1%81%D0%BB%D0%B0_%D0%9A%D0%B0%D1%82%D0%B0%D0%BB%D0%B0%D0%BD%D0%B0>  <http://e-maxx.ru/algo/catalan_numbers> |  |
| 46. Розрізування кола  хордами, що не перетиняються. | Скільки способів існує, щоб з'єднати 2N точок на колі N хордами, що не перетинячються. | Алгоритм розв’язання комбінаторних задач з використанням чисел Каталана | <http://e-maxx.ru/algo/catalan_numbers> |  |
| 47 Задача про черепашку | Дана прямокутна таблиця, в клітинах якої записані цілі числа. Черепашка знаходиться в лівій верхній клітині і їй необхідно потрапити в праву нижню клітину. За один хід Черепашка може переміститися в сусідню нижню або праву клітку. Потрібно знайти шлях Черепашки з максимальною сумою елементів. | Метод динамічного програмування | <http://kuimova.ucoz.ru/modul_8-dinamicheskoe_programmirovanie.pdf> |  |
| 48. Решето Ерастофена | Знайти всі прості числа на відрізку [1; n] за O (nloglog n) операцій та за лінійний час. | Метод динамічного програмування | <http://e-maxx.ru/algo/eratosthenes_sieve>  <http://e-maxx.ru/algo/prime_sieve_linear> |  |
| 49. Знаходження ступеня дільника факторіала | Дано два числа: N та K. Потрібно порахувати, з яким ступенем дільник K входить в число N!, тобто знайти найбільше X таке, що N! ділиться на K\*\*X. | Метод динамічного програмування | <http://e-maxx.ru/algo/factorial_divisors> |  |
| 50. Гра «Бики та корови» | В чисельній версії опонент задумує чотиризначне [число](https://uk.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE) (можливі варіанти з використанням числа будь-якої довжини). Всі цифри повинні бути різні. Тоді ж, в свою чергу, гравці намагаються вгадати число противника. Гравець пропонує свій варіант, а опонент дає кількість збігів. Якщо збігається цифра в її правильній позиції, то це є «бик», якщо не в своїй позиції — це «корова». | Метод динамічного програмування | <https://uk.wikipedia.org/wiki/%D0%91%D0%B8%D0%BA%D0%B8_%D1%82%D0%B0_%D0%BA%D0%BE%D1%80%D0%BE%D0%B2%D0%B8>  <https://habr.com/ru/post/148846/>  <http://slovesnov.users.sourceforge.net/bullscows/bullscows1_ru.pdf> |  |
| 51. Задача комівояжера | Існує n +1 міст A0, A1,…, An , n ≥ 1з заданими між ними відстанями dij (i, j = 0,1,...,n ). Потрібно, відправившись з A0 , вибрати такий маршрут пересування A0, Ai1,Аi2, Аі3,…, A0 , при якому комівояжер, побувавши в кожному місті по одному разу, повернувся б до вихідного пункту A0 , пройшовши при цьому мінімально можливий сумарний шлях | Метод динамічного програмування. Алгоритм Литтла | <https://er.knutd.edu.ua/bitstream/123456789/11822/1/NRMSE2018_V2_P491-492.pdf>  <http://proginschool.blogspot.com/2018/03/python.html> | Складність С |
| 52.Мишка і зернинки | Мишка і зернинки  В храмі підлогу прямокутної форми вимощено однаковими квадратними плитками 1 x 1, на кожну з яких насипано від 0 до k зернинок (k ≤ 30000). Розміри підлоги m х n. Мишка вибігає з лівого нижнього кута підлоги храму i рухається до входу у іншу нірку, розміщену у протилежному кутку. Мишка може рухатись лише праворуч або вперед, збираючи всі зернинки з плитки, на якій вона знаходиться. Знайти маршрут, рухаючись по якому мишка збере найбільшу кількість зернин. | Метод динамічного програмування |  |  |