

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ І ЗВ'ЯЗКУ**

Кафедра інженерії програмного забезпечення

**Створення багатомодульних
програмних проектів
для опрацювання даних у файлах**

Методичні рекомендації для виконання курсової роботи
з дисципліни

"Основи програмування"

для підготовки здобувачів першого (бакалаврського) рівня вищої
освіти у галузі знань «Інформаційні технології»
спеціальності Інженерія програмного забезпечення

Одеса
ДУІТЗ
2024

Укладачі:

Буката Л.М – старший викладач кафедри інженерії програмного забезпечення Державного університету інтелектуальних технологій і зв'язку.

Рецензенти:

Малахов Є.В., д.т.н., професор, зав. кафедри математичного забезпечення комп'ютерних систем Одеського національного університету ім. І.І. Мечникова.

Трегубова І.А., к.т.н., доцент, зав. кафедри комп'ютерних наук Державного університету інтелектуальних технологій і зв'язку.

*Рекомендовано навчально-методичною радою
Державного університету інтелектуальних технологій і зв'язку,
протокол № 4 від 05 березня 2024 р.*

Створення багатомодульних програмних проєктів для опрацювання даних у файлах : методичні рекомендації для виконання курсової роботи з дисципліни "Основи програмування". [Електронне видання] / уклад.: Буката Л.М. – Одеса: ДУІТЗ, 2024. – 39 с.

Методичні рекомендації містять матеріал для виконання курсової роботи з дисципліни "Основи програмування" та варіанти тематично підібраних завдань, які допоможуть засвоїти роботу зі створення багатомодульних програмних проєктів засобами мови програмування C++, закріпити навички з програмного опрацювання даних, які зберігаються у різнотипних файлах. Для самостійного виконання курсової роботи запропонована послідовність виконання цієї роботи на комп'ютері. Надано описи функцій виконання конкретних завдань. Рекомендації містять теоретичні відомості з докладним розглядом матеріалу.

Методичні рекомендації будуть корисними здобувачам вищої освіти за спеціальністю "Інженерія програмного забезпечення", які вивчають дисципліну "Основи програмування" для набуття навиків програмування з метою подальшого застосовування цих навиків у власній повсякденній і майбутній професійній діяльності; також стануть у нагоді користувачам персональних комп'ютерів, які бажають поглибити свої знання і навички з програмування засобами мови C++.

ЗМІСТ

ВСТУП	4
ЗАВДАННЯ ДО КУРСОВОЇ РОБОТИ.....	7
ВИМОГИ ДО СТРУКТУРНИХ ЕЛЕМЕНТІВ КУРСОВОЇ РОБОТИ.....	18
ВИМОГИ ДО ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ	20
ПОРЯДОК ПІДГОТОВКИ ТА ЗАХИСТ КУРСОВОЇ РОБОТИ	24
ПРИКЛАД ВИКОНАННЯ КУРСОВОЇ РОБОТИ	26
Завдання	26
Приклад опису функції first_word().....	26
Приклади схем алгоритмів роботи функцій	28
Програмний код у онлайн компіляторі C++ або у VS.....	28
Результати виконання програми.....	33
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	35
ДОДАТОК А Приклад оформлення титульної сторінки курсової роботи	36
ДОДАТОК Б Приклад завдання до курсової роботи.....	37

ВСТУП

Курсова робота «Створення багатомодульних програмних проєктів для опрацювання даних у файлах» з дисципліни Основи програмування є самостійною роботою, яка містить сукупність результатів алгоритмічних і програмних розробок, що свідчить про здібності здобувача проводити самостійні дослідження, використовуючи при цьому здобуті теоретичні знання та практичні навички.

У курсовій роботі неприпустимі порушення етики дослідження, серед яких: некоректні запозичення, порушення правил цитування, привласнення чужих наукових ідей, спотворення наукових фактів та ідей інших дослідників та результатів власного дослідження, джерел інформації тощо.

Вміст роботи може складатися з результатів теоретичних і практичних досліджень, застосуванні технологій та методичних прийомів при розв'язанні задач у галузі інформаційних технологій та програмування.

Виконання курсової роботи повинно свідчити про здатність здобувача самостійно здійснювати пошук, опрацьовувати та аналізувати інформацію з різних джерел, застосовувати знання і навички інформаційних технологій у практичних ситуаціях, вміти формулювати завдання дослідження, підбирати методи їх вирішення, вільно користуватись сучасними інформаційними технологіями для вирішення професійних завдань, програмувати і використовувати прикладні та спеціалізовані середовища для розв'язання різноманітних задач, інтерпретувати результати дослідження та формулювати висновки.

Виконання курсової роботи передбачає формування та розвиток у здобувачів вищої освіти таких загальних та спеціальних компетентностей[3]:

- здатність працювати в команді;
- Здатність до алгоритмічного та логічного мислення ;

та володіння інформаційними технологіями для вирішення професійних завдань, програмувати і використовувати прикладні та спеціалізовані комп'ютерно-інтегровані середовища для розв'язання різноманітних задач.

Внаслідок виконання курсової роботи здобувач вищої освіти буде здатним продемонструвати такі результати навчання:

- здійснювати професійну діяльність на основі знань сучасних інформаційно-комунікаційних технологій, працювати з програмним забезпеченням і файловою системою, обґрунтовано вибирати програмний засіб для вирішення фахових завдань і здійснювати обмін даними між програмами;
- застосовувати на практиці знання основ програмного забезпечення, різновидів алгоритмів та засобів їх побудови при розв'язуванні теоретичних і практичних задач;

- володіти термінологією проектування програм мовою програмування високого рівня (алфавіт мови, типи даних, терміни і правила синтаксису тощо), знаннями базових алгоритмів опрацювання даних та модульного принципу розробки програм;

- мати практичні навички роботи з сучасними платформами програмування для створення програм алгоритмічною мовою високого рівня;

- розробляти алгоритми різної структури та реалізовувати програми при розв'язанні теоретичних і практичних задач у галузі інформаційних технологій.

Результати навчання цієї курсової роботи деталізують такі програмні результати навчання:

- ПРН-1. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;

- ПРН-7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних й обчислювальних засобів інженерії програмного забезпечення.

- ПРН-13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

- ПРН-15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

У вказівках на конкретних прикладах розглянуто можливості мови С зі створення програмних проєктів з опрацювання даних, які зберігаються у файлах. Тематика роботи зумовлена тим, що більшість комп'ютерних програм, які вирішують ті чи інші практичні завдання, зберігають дані у вигляді різнотипних файлів, і тому виникає потреба програмно створювати, відкривати, читати, записувати файли. Мова С пропонує широкий спектр засобів зі створення бінарних (двійкових) і текстових файлів.

Актуальність теми курсової роботи "Створення багатомодульних програмних проєктів для опрацювання даних у файлах" зумовлена повсякденною практичною затребуваністю вмінь з розробки програмних проєктів, які зберігають і обробляють дані у файлах, тобто програма сама має створювати файли того чи іншого формату, записувати дані у файл, на вимогу виводити всі дані файлу, або лише ті, які задовольняють певній умові, вміти редагувати і видаляти певні дані файлу, надавати можливість створення нових файлів (звітних документів) з певними даними з основного файлу та ін.

Об'єктом дослідження є базові технології програмування алгоритмічною мовою високого рівня і засоби створення програмних проєктів.

Предмет дослідження – програмні засоби мови С для опрацювання різнорідних даних, створення файлів різних форматів, записування даних у файли, виве-

дення вмісту цих файлів, відбирання даних за умовою, можливості редагування даних у файлах, створення нових файлів (звітних документів) з певними даними з основного файлу.

Метою курсової роботи є:

а) систематизація, закріплення та розширення теоретичних і практичних знань за фахом та використання їх під час розв'язання конкретних прикладних задач у галузі інформаційних технологій;

б) розвинення навичок самостійної роботи з літературою, нормативними документами, пошуку необхідної інформації, оволодінні методикою дослідження й експериментування під час розв'язання задач і питань, які розроблюються у курсовій роботі;

в) закріплення знань алгоритмізації та навиків програмування алгоритмічною мовою високого рівня при опрацюванні різнорідних даних у файлах з можливістю їх програмного редагування;

г) виявлення підготовки здобувачів вищої освіти до самостійної роботи.

Відповідно до поставленої мети визначені такі задачі:

– розкриття сутності технологій мови С з опрацювання різнорідних даних у файлах;

– дослідження наявних сучасних програмних засобів, які дозволяють створювати програмні проєкти мовою С;

– вивчення засобів мови С з програмного створення файлів різних форматів, записування у них даних, перегляду вмісту цих файлів, відбору даних за умовою, редагування даних у файлах (для задач підвищеного рівня складності);

– розроблення програмного проєкту для вирішення завдання відповідно до індивідуального варіанта, який передбачає створення файлів, заповнення їх даними, відбір та редагування файлів різних форматів.

Курсова робота це самостійна робота здобувачів вищої освіти, яка передбачає закріплення та розширення теоретичних знань з дисципліни "Основи програмування".

Дані методичні рекомендації визначають загальні вимоги до курсової роботи, її приблизну тематику та структуру, правила оформлення та порядок підготовки роботи до захисту.

ЗАВДАННЯ ДО КУРСОВОЇ РОБОТИ

Використовуючи мову програмування C, створити програмний проєкт, який реалізує такі завдання з опрацювання різнорідних даних, що зберігаються у файлах:

1) як вхідні дані для програмного опрацювання файлових даних створити текстовий файл, наприклад, з ім'ям *F1*, в який записати числові дані у вигляді матриці розмірності M на N (M – кількість рядків, N – кількість стовпців);

2) у програмному проєкті розробити функцію, яка надасть можливість вивести на екран вміст текстового файла *F1*, ім'я якого передаватиметься як аргумент із функції `main()`;

3) створити функції для:

а) формування числової матриці розмірності M на N зчитуванням її значень із файла *F1*. Для базового та підвищеного рівня (*) розмірність матриці M та N задати константами, а для високого рівня (**) розмірність матриці є довільною і має визначатися програмно;

б) виведення цієї матриці на екран та у файл *F2*;

4) створити функцію для опрацювання числової матриці з обчисленням значень відповідно до індивідуального варіанта (завдання а) та б) (*) у табл. 1). Результати обчислень записати у файл *F2*;

5) створити функцію для опрацювання числової матриці з формуванням елементів вектора відповідно до індивідуального варіанта (завдання в у табл. 1). Результати обчислень записати у файл *F2*;

6) * створити одновимірний динамічний масив з елементів матриці за заданою умовою відповідно до індивідуального варіанта (завдання г у табл. 1). Результати обчислень записати у файл *F2*;

7) ** створити двовимірний динамічний масив з елементів матриці за заданою умовою відповідно до індивідуального варіанта (завдання д у табл. 1). Результати обчислень записати у файл *F2*;

8) як вхідні дані для програмного опрацювання текстових даних створити ще один файл з ім'ям *F3.txt* та заповнити його коректними текстовими даними;

9) програмно переглянути на екрані вміст текстового файла *F3.txt*, ім'я якого передаватиметься як аргумент із функції `main()`;

10) розробити функцію, яка виконуватиме програмне опрацювання даних з файла *F3.txt*, відповідно до завдання у таблицях 2...3, згідно з індивідуальним варіантом. Програмно записати результати опрацювання текстових даних у файл (або файл *F2.txt*, або новий). Програмно переглянути на екрані вміст текстового файла з результатами опрацювання текстових даних;

11) відредагувати файл *F3.txt* з текстовими даними, після чого переглянути на екрані його вміст:

а)* рядок, заданий номером (номер ввести з клавіатури);

б)** рядки, які містять заданий текст (текст ввести з клавіатури);

12) видалити з файла *F3.txt*, після чого переглянути на екрані його вміст:

- а)* рядок, заданий номером;
- б)** рядки, які містять заданий текст.

Індивідуальні варіанти завдань надані у табл. 1–3.

Для виконання завдання пропонується три рівня складності: базовий, підвищений (*) та високий (**). Для базового рівня достатньо виконати п. 1–5 та 8–10, а для підвищеного – ще пункти завдання, позначені однією зірочкою (*). Високий рівень складності, крім виконання завдань базового та підвищеного рівнів, передбачає виконання завдань, позначених двома зірочками (**), та винесення усіх функцій у власну бібліотеку (статичну або динамічну). Також можливе виконання нових (не розглянутих у базовому курсі навчання) програмних інструментів і технологій.

Для всіх функцій записати словесні описи та схеми алгоритму.

Таблиця 1

Індивідуальні завдання

Варіант	Завдання
1	а) Обчислити значення середнього арифметичного елементів останнього рядка матриці. - б)* Знайти максимальне з чисел, що зустрічаються в заданій матриці більше одного разу. в) Обчислити елементи вектору як елементи рядків матриці, значення яких більше за 20. - г)* Сформувати одновимірний динамічний масив із елементів, які перевищують середнє арифметичне усіх елементів матриці. - д)** Сформувати нову матрицю з показників тих датчиків (рядків матриці), які містять значення вимірювань лише в діапазоні [50, 100].
2	а) Обчислити кількість нульових елементів матриці. - б)* Визначити кількість стовпців, які не містять жодного нульового елемента. в) Обчислити елементи вектору як максимальні значення кожного рядка матриці. - г)* Сформувати одновимірний масив з чисел, розміщених до першого нульового числа матриці (якщо нульових чисел немає, вибрати всі). - д)** Характеристикою рядка матриці назвемо суму її додатних елементів. Переставляючи рядки заданої матриці, розташувати їх відповідно до зростання характеристик.

Продовження табл. 1

Варіант	Завдання
3	а) Обчислити мінімальне та максимальне значення елементів матриці. б) * Знайти номер рядка, в якому є найдовша серія однакових елементів. в) Обчислити елементи вектора як кількості елементів кожного з елементів у рядках, які перевищують 30. г) * Сформувати одновимірний динамічний масив з чисел, розміщених після максимального значення останнього рядка матриці. д) ** Сформувати нову матрицю з елементів тих рядків матриці, які не містять жодного нульового значення.
4	а) Обчислити значення кількості елементів матриці, абсолютне значення яких менше 1. б) * Визначити кількість ненульових елементів лише у тих рядках, які не містять від'ємних елементів. в) Обчислити елементи вектора як середнє арифметичне елементів першого та останнього рядків матриці. г) * Сформувати одновимірний масив з чисел, розміщених між максимальним і мінімальним значеннями першого рядка. У разі, коли ці елементи розташовані поруч, вибрати у вектор увесь рядок. д) ** Сформувати нову матрицю із елементів тих стовпців матриці, які мають елементи більші за 25.
5	а) Визначити значення максимального значення другого рядка матриці. б) * Визначити загальну суму елементів матриці лише в тих стовпцях, які не містять від'ємних елементів. в) Обчислити елементи вектора як середньо-арифметичні значення тих елементів рядків, абсолютне значення яких не перевищує 5. г) * Створити динамічний масив із чисел, розміщених до першого трицифрового числа (якщо трицифрових чисел немає, вибрати всі). д) ** Створити нову матрицю з тих стовпців матриці, які не містять елементи з діапазону (10, 20].
6	а) Обчислити кількість показників елементів матриці зі значенням, меншим за 15. б) * Обчислити суму елементів першого з рядків матриці, який не містить від'ємних чисел. в) Обчислити елементи вектора як добутки відповідних елементів другого та останнього стовпців матриці. г) * Обчислити вектор як середньо арифметичні значення елементів тих стовпців, які містять максимальний і мінімальний елементи. д) ** Упорядкувати рядки матриці за збільшенням кількості однакових елементів у кожному рядку.

Варіант	Завдання
7	а) Обчислити середнє арифметичне значення елементів другого рядка матриці. б) * Знайти загальну суму елементів у тих стовпцях, які містять хоча б один від'ємний елемент. в) Визначити елементи вектора як рядок матриці з максимальним елементом. г) * Створити динамічний масив з чисел, розміщених до першого елемента, більшого за 10. д) ** Створити нову матрицю з тих рядків матриці, які не містять жодного елемента 0 чи понад 127.
8	а) Обчислити кількість елементів матриці, більших за 1. б) * Елемент матриці називається локальним мінімумом, якщо він менше всіх його сусідів (ліворуч, праворуч, зверху і знизу). Підрахувати кількість локальних мінімумів заданої матриці. в) Обчислити вектор як суми від'ємних елементів стовпців матриці. г) * Створити динамічний масив із ненульових елементів усіх рядків. д) ** Характеристикою рядка матриці назвемо квадрат суми його елементів. Переставляючи рядки заданої матриці, розташувати їх відповідно до зростання характеристик.
9	а) Обчислити значення кількості елементів зі значенням, більшим за середнє арифметичне елементів матриці. б) * Визначити суму усіх елементів у тих рядках, які містять хоча б один від'ємний елемент. в) Обчислити елементи вектора як різниці відповідних елементів першого і третього стовпців матриці. г) * Створити динамічний масив із чисел, розміщених після останнього елемента зі значенням нуль для останнього рядка (якщо нулів немає, вибрати всі числа рядка). д) ** Створити нову матрицю з тих рядків, у яких є сідлові точки матриці. <i>Примітка.</i> Сідловою точкою цієї матриці назвемо елемент, який водночас є мінімумом у своєму рядку і максимумом у своєму стовпці.
10	а) Обчислити значення мінімального додатного елемента матриці. б) * Знайти суму елементів у тих стовпцях, які містять хоча б один від'ємний елемент. в) Обчислити елементи вектора як середнє арифметичне додатних елементів рядків матриці. г) * Створити динамічний масив лише з чисел, модуль яких не попадає в проміжок (10, 50]. д) ** Створити нову матрицю з тих рядків матриці, які не мають значення елементів, більших за 100.

Продовження табл. 1

Варіант	Завдання
11	а) Обчислити кількість елементів, значення яких більше за значення першого елемента матриці. б) * Обчислити середнє арифметичне елементів, розміщених до найбільшого елемента. в) Обчислити вектор квадратів елементів мінімальних показників кожного стовпця матриці. г) * Створити динамічний масив лише з чисел, які завершуються цифрою 1 у цілій частині числа. д) ** Створити нову матрицю з тих рядків матриці, які не мають елементів, за модулем менших за 9.
12	а) Обчислити суму модулів від'ємних елементів матриці. б) * Обчислити суму елементів того рядка матриці, який не містить нульових значень. Якщо таких рядків декілька, взяти їх усі. в) Обчислити елементи вектора як середньо арифметичні значення елементів рядків матриці. г) * Створити динамічний масив з елементів першого з рядків матриці, який не містить від'ємних чисел. д) ** Створити нову матрицю з тих рядків матриці, які не містять елементи, дійсна частина яких дорівнює 0.
13	е) Обчислити добуток одноцифрових елементів матриці. ж) * Обчислити середнє арифметичне елементів того рядка матриці, який містить максимальний елемент. з) Обчислити вектор квадратів значень останніх елементів рядків матриці. и) * Створити динамічний масив з елементів матриці за винятком нульових і від'ємних чисел. к) ** Створити нову матрицю, в якій для кожного рядка в окремому рядку матриці, розміщеному нижче під значеннями відповідного рядка, обчислити різниці від середньо арифметичного цього рядка.
14	а) Обчислити кількість додатних елементів і кількість нульових елементів матриці. б) * Обчислити середнє арифметичне значення стовпця матриці, який містить мінімальне значення матриці. У разі декількох однакових мінімальних значень, брати перший з них. в) Обчислити елементи вектора як суми елементів рядків матриці з парними індексами стовпців. г) * Створити динамічний масив з чисел, розміщених після першого двоцифрового числа матриці. д) ** Ущільнити задану матрицю, видаляючи з неї рядки і стовпці, заповнені нулями, тобто сформувати нову відповідну матрицю.

Варіант	Завдання
15	а) Обчислити значення добутку показників останнього рядка матриці. б) * Визначити номер першого зі стовпців, що містять хоча б один нульовий елемент. в) Обчислити елементи вектора як стовпець матриці з максимальним елементом. У разі декількох однакових максимальних значень, брати останній з них. г) * Створити динамічний масив із елементів матриці, які завершуються цифрою 5 у цілій частині. д) ** Характеристикою рядка матриці назвемо суму її від'ємних елементів. Переставляючи рядки заданої матриці, розташувати їх відповідно до спадання характеристик.
16	а) Обчислити кількість нульових елементів матриці. б) * Обчислити середнє арифметичне значення усіх максимальних елементів рядків матриці. в) Обчислити вектор сум додатних елементів рядків матриці. г) * Створити масив з показників, розміщених після першого числа, більшого за 100 (якщо таких чисел немає, вибрати всі). д) ** Створити нову матрицю з тих рядків матриці, які не містять від'ємних елементів.
17	а) Обчислити середнє арифметичне значення елементів матриці з парними індексами. б) * Знайти номер першого зі стовпців, що не містять жодного від'ємного елемента. в) Обчислити елементи вектора як максимальні елементи рядків матриці. г) * Створити масив з елементів, модуль яких попадає у проміжок [5, 50). д) ** Створити нову матрицю з тих стовпців матриці, які містять хоча б одне нульове значення.
18	а) Обчислити значення найменшого елемента матриці. б) * Визначити кількість рядків, що мають хоча б одне нульове значення вимірювань. в) Обчислити елементи вектора як суми максимального і мінімального елементів рядків матриці. г) * Створити динамічний масив з чисел, розміщених до першого від'ємного числа (якщо від'ємних двоцифрових чисел немає, вибрати всі). д) ** Створити нову матрицю з тих стовпців матриці, які не містять елементи менші 10.

Продовження табл. 1

Варіант	Завдання
19	а) Обчислити значення суми додатних елементів матриці. б) * Визначити номер стовпця, в якому є щонайдовша серія однакових елементів. в) Обчислити елементи вектора як середнє арифметичне елементів рядків матриці. г) * Створити динамічний масив з чисел, розміщених після першого нульового значення (якщо нульових чисел немає, вибрати всі). д) Створити нову матрицю з тих показників датчиків (рядків матриці), які містять елементи лише в діапазоні [50, 100].
20	а) Обчислити різницю між абсолютними значеннями максимального і мінімального елементів матриці. б) * Визначити суму елементів у тих рядках, які не містять від'ємних елементів. в) Обчислити елементи вектора як суми елементів стовпців матриці. г) * Створити динамічний масив із елементів, які мають серед своїх цифр хоча б одну цифру 1. д) ** Створити нову матрицю з тих елементів рядків матриці, які містять хоча б один нульовий елемент.
21	а) Обчислити значення добутку елементів останнього вимірювання (стовпця матриці). б) * Визначити мінімум серед сум елементів рядків матриці. в) Обчислити елементи вектора як середньо арифметичні значення елементів першого та останнього рядків. г) * Створити динамічний масив із елементів, які мають значення, більші за середнє арифметичне усіх елементів. д) ** Шляхом перестановки рядків матриці домогтися того, щоб її мінімальний елемент був розміщений у першому рядку, наступний за величиною – у другому, подальший за величиною – у третьому і т. д.
22	а) Обчислити значення кількості від'ємних елементів для перших трьох рядків матриці. б) * Визначити добуток додатних елементів тих рядків, які містять хоча б один нульовий елемент. в) Обчислити елементи вектора як найбільші за модулем елементи рядків. г) * Створити динамічний масив з елементів матриці, які відрізняються від числа 5 не більше ніж на 10. д) ** Ущільнити матрицю, видаляючи з неї рядки і стовпці, які містять від'ємні елементи.

Варіант	Завдання
23	а) Обчислити середнє арифметичне значення елементів четвертого рядка матриці. б) * Знайти суму значень елементів матриці, які зустрічаються понад двічі. в) Обчислити елементи вектора як суму відповідних елементів першого та третього стовпців матриці. г) * Створити динамічний масив з елементів матриці у тих рядках, які містять хоча б один 0. д) ** Упорядкувати рядки матриці за збільшенням кількості однакових елементів у кожному рядку.
24	а) Обчислити модуль суми всіх від'ємних значень елементів матриці. б) * Знайти загальну суму елементів рядків, в яких однакові перший та останній елементи. в) Обчислити елементи вектора як різницю максимального та мінімального елементів рядків матриці. г) * Створити динамічний масив з чисел матриці, розміщених після першого числа зі значенням 1 (якщо таких чисел немає, вибрати показники тільки останнього рядка). Обхід матриці здійснювати за стовпцями. д) ** Створити нову матрицю з тих стовпців матриці, які не містять жодного значення 10.
25	а) Обчислити кількість додатних елементів матриці. б) * Знайти добуток елементів стовпця (або стовпців), який містить максимальний елемент. в) Обчислити елементи вектора як стовпець з мінімальною сумою елементів. У разі декількох однакових максимальних значень, брати останній зі стовпців. г) * Створити динамічний масив лише з додатних чисел матриці першого та останнього рядків. д) ** Створити нову матрицю з тих рядків матриці, які містять хоча б один елемент – двійку.
26	а) Обчислити середнє арифметичне елементів матриці, значення яких належать проміжку [10, 20]. б) * Знайти кількість стовпців, які не містять від'ємних елементів. в) Обчислити елементи вектора як квадрати значень останніх елементів стовпців матриці. г) * Створити динамічний масив лише з ненульових показників останніх трьох рядків. д) ** Створити нову матрицю з тих стовпців матриці, які містять хоча б один елемент 2 або -1.

Закінчення табл. 1

Варіант	Завдання
27	а) Обчислити мінімальний за модулем елемент матриці. б) * Обчислити номер рядка, довжина якого (як вектора) є максимальною. в) Обчислити вектор як модулі сум елементів рядків матриці. г) * Сформувати одновимірний масив з чисел, розміщених між першим максимальним і останнім мінімальним елементами матриці. Обхід матриці здійснюється за рядками. д) ** Перестановкою рядків упорядкувати матрицю за зменшенням елементів першого стовпця матриці.
28	е) Обчислити суму елементів матриці зі значенням більшим за середнє арифметичне усіх елементів. ж) * Обчислити номер стовпця матриці з мінімальною сумою елементів. Якщо таких стовпців декілька, взяти перший із них. з) Обчислити елементи вектора як квадрати сум від'ємних елементів стовпців матриці. и) * Створити динамічний масив зі значень, розміщених до першого від'ємного числа матриці. Обхід матриці здійснювати по стовпцях зверху вниз. к) ** Характеристикою рядка матриці назвемо добуток її парних елементів. Переставляючи рядки заданої матриці, розташувати їх відповідно до спадання характеристик.
29	л) Обчислити середнє арифметичне значення максимального і мінімального елементів матриці. м) * Обчислити елементи вектора як скалярний добуток елементів рядків матриці на останній стовпець. н) Обчислити вектор як більший із елементів першого та останнього рядків матриці. о) * Сформувати одновимірний масив з чисел, розміщених між першим та останнім нульовими елементами матриці. п) ** Переходом назвемо переміщення до сусіднього елемента по горизонталі або по вертикалі. Знайти кількість різних шляхів проходу матриці з лівого верхнього кута в правий нижній, якщо допустимими є тільки переходи "праворуч на один елемент" і "вниз на один елемент". Визначити шлях, сума елементів якого максимальна.
30	а) Обчислити значення кількості додатних елементів матриці. б) * Віднайти значення мінімального додатного елемента матриці. в) Обчислити елементи вектора як кількості від'ємних елементів рядків. г) * Сформувати одновимірний масив з чисел, розміщених між першим та другим нульовими елементами матриці. д) ** Характеристикою рядка матриці назвемо суму її додатних елементів. Переставляючи рядки заданої матриці, розташувати їх відповідно до зростання характеристик.

Таблиця 2

Варіант	Завдання базового рівня
1	Вивести усі розділові знаки та їхні індекси у кожному рядку
2	Визначити кількість слів у рядку. Слова розділяються одним чи декількома пробілами
3	Створити новий рядок із розділових знаків рядків
4	Створити нові рядки із малих літер латиниці рядків файла
5	Визначити, чи є в рядках літера X (маленька або велика)
6	Видалити у рядках усі пробіли
7	Створити нові рядки з останніх літер кожного зі слів у рядках файла
8	Визначити, чи є у рядках цифри. Якщо так, то вивести їх
9	Визначити кількість слів, після яких є розділові знаки, та вивести ці знаки
10	Створити нові рядки з великих літер кожного рядка
11	Визначити кількість зайвих пробілів у кожному рядку (подвійних, початкових і кінцевих)
12	Вивести індекси всіх ком та крапок, які є у рядках
13	Вивести порядкові номери слів введеного рядка, які містять цифри
14	Перевести останні зі слів у рядках до нижнього регістра
15	Визначити, чи є у кожному рядку знаки питання або оклику
16	Із рядків створити нові рядки за винятком знаків пунктуації
17	Вивести індекси всіх цифр, які є у рядку
18	Вивести останні літери усіх слів рядка
19	Перетворити регістр літер рядків: якщо перша літера рядка є великою (верхній регістр) перевести до верхнього регістра весь рядок, і навпаки
20	Визначити кількість слів рядка, які починаються з літер латиниці
21	Вивести індекси всіх входжень літери C до рядка
22	Створити новий рядок з цифр, які містяться у рядку
23	Вивести перші літери усіх слів рядка
24	Створити новий рядок з першого й останнього слів рядка
25	Створити новий рядок за винятком першого слова рядка
26	Визначити, чи є у рядках знаки арифметичних операцій. Якщо так, то створити новий рядок із цих символів
27	Вивести останнє слово рядка
28	Вивести перше слово введеного рядка у зворотному порядку
29	Знайти індекси початку й кінця другого слова у рядку
30	Відібрати рядки, що з обох боків оточені цифрами

Таблиця 3

Варіант	Завдання підвищеного рівня
1	Відсортувати слова за абеткою у кожному рядку
2	Переставити всі знаки пунктуації у кінець рядка
3	Знайти у кожному рядку найдовшу послідовність літер, що розташовані за абеткою
4	Вивести рядки, які містять хоча б одне слово лише з великих літер
5	Вивести слова файла, які містять щонайменше дві літери 'а', та обчислити їхню кількість
6	Змінити порядок символів у рядку на зворотній, не використовуючи інших рядків
7	Вивести слова текстового файла, які закінчуються на голосну літеру, та обчислити їхню кількість
8	Видалити з рядка їхні «серединки», залишивши перше та останнє слова у ньому
9	Замінити числа, незалежно від кількості цифр у них, на літеру Х
10	Поміняти місцями перше й останнє слова кожного рядка у файлі
11	Вивести кількість приголосних у кожному рядку
12	Вивести слова файла, які розпочинаються й закінчуються на одну й ту саму літеру
13	Вивести найдовше слово у кожному рядку
14	Переставити навпаки символи останнього слова кожного рядка
15	Відсортувати за абеткою символи першого слова рядка
16	Вивести довжину найдовшого слова у кожному рядку
17	Для кожного рядка вивести літери, які зустрічаються декілька разів
18	Підрахувати кількість голосних у рядках, що з обох боків оточені приголосними
19	Перенести усі пробіли кожного рядка на його початок
20	Замінити усі голосні літери кожного рядка на першу голосну літеру у цьому рядку
21	Вивести рядок файла із найдовшою послідовністю цифр
22	Обчислити середню довжину слів у кожному рядку
23	Вивести слова файла, які не містять літеру 'о', та визначити їхню кількість
24	Вивести рядок текстового файла, який містить найдовше слово
25	Вивести слова файла, після яких є кома, та визначити їхню кількість
26	Вивести найкоротше слово у кожному рядку
27	Вивести слова файла з довжиною більше семи символів та обчислити їхню кількість
28	Вивести всі слова, які починаються з літери 'ф' чи 'Ф', обчислити їхню кількість
29	Видалити з кожного рядка слова з довжиною, менше трьох символів
30	Вивести рядок файла із найменшою послідовністю цифр

ВИМОГИ ДО СТРУКТУРНИХ ЕЛЕМЕНТІВ КУРСОВОЇ РОБОТИ

Текст курсової роботи може мати такі складові наповнення:

- 1) титульна сторінка;
- 2) завдання із зазначенням індивідуального варіанта;
- 3) зміст;
- 4) вступ;
- 5) розділ 1 "Аналіз засобів та методів розробки";
- 6) розділ 2 "Опис алгоритмів розв'язування задач";
- 7) розділ 3 "Результати виконання програми";
- 8) розділ 4 "Інструкція з використання програми";
- 9) висновки з аналізом досягнутих результатів;
- 10) список використаних джерел;
- 11) додатки.

Оптимальний обсяг основної частини курсової роботи (без додатків) – 25 – 30 друкованих сторінок.

Курсова робота починається з *титульної сторінки* за формою, наданою у додатку А. Це перша сторінка курсової роботи, яку включають у загальну нумерацію сторінок, але не нумерують. Далі номер сторінки проставляють у правому верхньому куті.

За титульною сторінкою розміщують завдання до курсової роботи з календарним планом виконання, приклад якого надано у додатку Б.

Далі у пояснювальній записці курсової роботи розміщують послідовно зміст, вступ, розділи у порядку подання, висновки, список використаних джерел, додатки. Кожну структурну частину роботи починають з нової сторінки. Заголовки структурних частин роботи друкують великими літерами симетрично відносно тексту.

Зміст має бути створений з можливістю автоматичного оновлення заголовків структурних частин роботи та нумерацією сторінок. Для цього у тексті для форматування треба застосовувати стильове форматування (див. Вимоги до оформлення курсової роботи).

У **вступі** слід зазначити предмет, об'єкт та мету дослідження, сформулювати завдання, за допомогою якого може бути досягнута мета дослідження, розкрити актуальність тематики роботи, тобто причини важливості дослідження вибраної тематики.

Крім того, у вступі слід викласти структуру роботи, наприклад: "Курсова робота складається зі вступу, чотирьох розділів, висновків та списку використаних джерел. Загальний обсяг курсової роботи: 28 с., 7 рис., 4 табл., 2 додатки, 14 джерел."

Основну частину курсової роботи надають, поділяючи матеріал на розділи. Основна частина роботи складається з трьох – чотирьох розділів.

У **розділі 1** основної частини розглядаються теоретичні та методологічні аспекти досліджуваної задачі, аналітичний огляд літературних джерел з предмета наукового дослідження, огляд методів, які використовуються для розв'язання задачі, теоретичне обґрунтування, суть, значення, класифікаційні характеристики та тенденції розвитку предмета дослідження тощо. Для констатації та обґрунтування загальнотеоретичних тенденцій доцільно використовувати дані, опубліковані у відповідних джерелах та енциклопедіях, монографіях, довідниках, зарубіжних джерелах та виданнях[1]. У цьому розділі доцільно сформулювати опис тих засобів, які використовувалися при написанні програмного коду для виконання завдання. Цей розділ є теоретичним, у ньому бажано розкрити сутність досліджуваних процесів і використовуваних методів для розкриття сутності зазначеної роботи, може надаватися їхня класифікація та різні трактування з позицій різних авторів у науковій літературі.

Розділ 2 містить докладний словесний опис алгоритмів усіх функцій, розроблених автором, для розв'язування індивідуальних задач.

Розділ 3 "Результати виконання програми" висвітлює розробки автора і містить результати досліджень. Цей розділ носить практичний характер і містить ілюстрацію (скріншоти) результатів виконання програмного проекту.

Розділ 4 присвячується детальному опису інструкції щодо використання розробленої програми. Можна надати конкретні обґрунтовані пропозиції щодо використання розробленої програми.

Висновки розміщують безпосередньо після викладення тексту, починаючи з нової сторінки. У висновках надаються підсумки проведеного дослідження, слід зробити всебічний аналіз отриманих результатів та набутих знань і навичок (при цьому заохочується самостійність суджень та оцінок). Формулювання висновків повинно базуватися на матеріалах основної частини роботи відповідно до поставлених завдань. У даному пункті автор курсової роботи оцінює повноту виконання поставлених завдань та формулює рекомендації щодо можливостей практичного використання матеріалів роботи.

Перелік використаних джерел має бути оформлений за вимогами ДСТУ 8302:2015 «Інформація та документація [10]. Бібліографічне посилання. Загальні положення та правила складання» і містити 10 – 20 літературних джерел, включаючи посилання на інтернет-ресурси. Тут надають літературу, на які в тексті є посилання, а також ту, яку використано при викладенні конкретних теоретичних положень. Джерелами інформації можуть бути сучасні навчальні та навчально-методичні матеріали, статті, монографії, нормативно-технічні документи, інформаційні інтернет-ресурси тощо. Перелік джерел посилання подається мовою оригіналу в порядку згадування джерела у роботі, а самі посилання в тексті нумеруються послідовно арабськими числами і подаються у квадратних дужках: [1], [2–6], [7, 8]. На всі літературні джерела, надані в переліку використаних джерел, посилання повинні бути в тексті роботи обов'язково.

Додатки нумерують прописними літерами українського алфавіту по центру, наприклад, «Додаток А». Найменування додатка прописують рядком нижче прописними літерами по центру. Додатки можуть містити допоміжні матеріали: схеми алгоритмів усіх функцій, тексти розроблених програм, таблиці, ілюстрації допоміжного характеру тощо. Наповнення додатків курсової роботи може бути таким:

Додаток А «Структурні схеми алгоритмів функцій проекту». Тут надаються розроблені автором блок-схеми (графічне подання алгоритмів) для усіх функцій для розв'язування локальних задач. Алгоритми подають у вигляді рисунків, з обов'язковою нумерацією за номером додатка та з наданням назви рисунка.

Додаток Б «Програмний код проекту» містить лістинги з текстом програмного проекту мовою програмування високого рівня усіх модулів із докладними коментарями і поясненнями.

ВИМОГИ ДО ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ

Оформлення пояснювальної записки курсової роботи має відповідати загальним вимогам до наукових робіт згідно з державним стандартом ДСТУ 3008-2015 «Документація. Звіти у сфері науки та техніки. Структура і правила оформлення»[10]. Виконана курсова робота здається викладачеві у прошитому вигляді або у спеціальних (пластикових) теках.

Текст роботи оформляється на комп'ютері у текстовому редакторі на аркушах формату А4 (210х297 мм). Встановлюються такі поля сторінок: ліве – 2,5 см, праве – 1 см, верхнє і нижнє – 2 см.

Для форматування тексту курсової роботи варто застосовувати стильове форматування:

- стиль *Звичайний* – для основного тексту. Його налаштування:
 - ✓ шрифт: Times New Roman, розмір – 14 пт, колір – чорний, щільність тексту – звичайна;
 - ✓ параметри абзацу: відступ першого рядка абзацу 1,25 см; додаткові інтервали між абзацами, окрім встановленого міжрядкового інтервалу, не встановлюються (відступи перед і після – 0, інтервал перед і після – 0); міжрядковий інтервал «Множинний» – 1,25 пт; вирівнювання – за шириною;
- стиль *Заголовок1* застосовувати лише до всіх назв основних структурних частин роботи. Його налаштування:
 - ✓ шрифт: Times New Roman, розмір – 14 пт, колір – чорний, жирний; усі великі букви;
 - ✓ параметри абзацу: відступ першого рядка абзацу 0 см; міжрядковий інтервал «Множинний» – 1,25 пт; вирівнювання – по центру;
- стиль *Заголовок2* застосовувати до назв підрозділів. Його налаштування:
 - ✓ шрифт: Times New Roman, розмір – 14 пт, колір – чорний, жирний;

- ✓ параметри абзацу: відступ першого рядка абзацу 1,25 см; міжрядковий інтервал «Множинний» – 1,25 пт; вирівнювання – за лівим краєм.

Таблиці оформляються шрифтом 12 пт (за потреби 10 пт).

Уся текстова частина курсової роботи оформляється не жирним шрифтом, без курсиву, винятком є заголовки та підзаголовки. У роботі не допускається виділення фрагментів основного тексту іншими шрифтами, кольорами, підкреслюванням тощо.

Заголовки основних структурних частин курсової роботи розміщують з нової сторінки. Їх друкують прописними літерами жирним шрифтом по центру сторінки відповідно до стилю *Заголовок1*. Номер розділу ставлять перед назвою розділу, слово «РОЗДІЛ» не пишеться, після номера крапку не ставлять, наприклад:

1 АНАЛІЗ ЗАСОБІВ ТА МЕТОДІВ РОЗРОБКИ

Наприкінці заголовків крапки не ставлять. Якщо заголовок складається з двох або більше речень, їх розділяють крапкою.

Відстань між заголовком розділу та підрозділу – 2 порожні рядки.

Підзаголовки повинні бути виділені жирним шрифтом, вирівнювання – за лівим краєм відповідно до стилю *Заголовок2*. Підрозділи нумеруються у межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, між ними ставиться крапка, наприклад: "1.2" (другий підрозділ першого розділу). У тому самому рядку пишеться назва підрозділу малими літерами (крім першої прописної), наприклад:

1.2 Засоби розроблення проєкту

Відстань між заголовком підрозділу і подальшим текстом (назвою пункту) або попереднім текстом має бути виокремлено одним порожнім рядком.

Підрозділи можуть містити пункти, які нумеруються у межах певного підрозділу. Номер пункту складається з номера підрозділу і порядкового номера пункту, між ними ставиться крапка, наприклад: "1.2.1" (перший пункт другого підрозділу у першому розділі).

Відстань між заголовком пункту або підпункту і подальшим текстом відсутня. Але між попереднім текстом та пунктом має бути виокремлено одним порожнім рядком.

Відстань між рисунком, таблицею, формулою (разом з підписом до рисунка чи табличною назвою) і подальшим або попереднім текстом має бути виокремлено одним порожнім рядком.

Нумерацію сторінок, розділів, підрозділів, додатків, рисунків, таблиць подають арабськими цифрами без знаку «№». Нумерація сторінок повинна бути наскрізною та проставлятися арабськими цифрами у правому верхньому куті аркуша. Розмір шрифту нумерації – 12 пт.

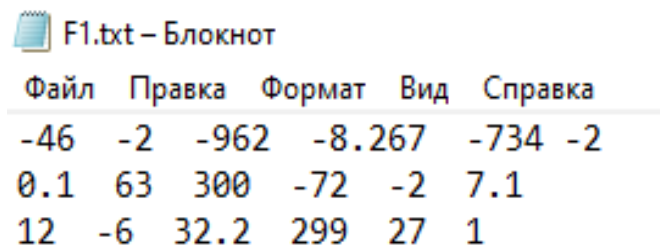
Графічні ілюстративні матеріали позначають словом «Рисунок» та нумерують послідовно в межах розділу. Номер, назву рисунка та пояснювальний підпис

розміщують послідовно, відокремлюючи тире, наприклад: Рисунок 2.1 – (перший рисунок другого розділу), за винятком рисунків, наданих у додатках. Підпис рисунка розміщують по центру під рисунком, крапка в кінці не ставиться. Рисунок та його підпис повинні бути розміщені на одній сторінці. Рисунки необхідно розміщувати безпосередньо після тексту, де вони згадані вперше, або на наступній сторінці.

На всі надані рисунки в тексті курсової роботи повинні бути посилання або в дужках (рис. 3.2), або по контексту, наприклад, «... як показано на рис. 3.2». Допускається вертикальне розміщення рисунків за годинниковою стрілкою. Рисунок, розмір якого більший формату А4, рекомендується розміщувати у додатках.

Рисунки відокремлюються від основного тексту зверху та знизу одним порожнім рядком.

Приклад рисунка з відповідним підписом:



Файл	Правка	Формат	Вид	Справка
-46	-2	-962	-8.267	-734 -2
0.1	63	300	-72	-2 7.1
12	-6	32.2	299	27 1

Рисунок 3.2 – Вигляд текстового файла F1.txt з вхідною матрицею числових елементів

Так само і таблиці відокремлюються від основного тексту зверху і знизу одним порожнім рядком. Таблицю слід розташовувати безпосередньо після тексту, в якому вона згадується вперше, або на наступній сторінці. При перенесенні таблиці на наступну сторінку тематичний заголовок не повторюють, а пишуть «Продовження табл. 1.1» або «Закінчення табл. 1.1» та повторюють нумерацію на наступній сторінці.

На всі таблиці мають бути посилання в тексті. При цьому по тексту слово «таблиця» пишуть скорочено, наприклад: «...у табл. 2.1». У повторних посиланнях на таблиці та ілюстрації скорочено пишуть слово «дивись», наприклад: «див. табл. 2.1».

Кожна таблиця повинна мати тематичний заголовок з абзацним відступом. Назву таблиці друкують нежирним шрифтом малими літерами (крім першої великої). Якщо числові чи інші дані в деяких комірках таблиці відсутні, у ньому ставлять прочерки. Числа у комірках таблиці варто проставляти так, щоб відповідні розряди чисел в усьому стовпчику були розташовані один під одним. Таблиця оформлюється за шириною сторінки.

Кожну таблицю варто супроводжувати коротким аналізом чи коментарем.
Приклад оформлення таблиці:

Таблиця 2.1 – Специфікатори режиму відкривання файлів

Параметр	Опис
r	відкрити файл лише для зчитування даних
r+	відкрити файл для зчитування й записування даних

Закінчення табл. 2.1

Параметр	Опис
a	відкрити чи створити файл для записування даних у кінець файла
a+	відкрити чи створити файл для зчитування чи то записування даних у кінець файла
w	створити файл для записування даних
w+	створити файл для зчитування і записування даних

Нумерація таблиць і рисунків, розміщених в додатках, нумерують так:

- таблиці – по центру над таблицею «Таблиця А.1 – Назва таблиці»;
- рисунки – по центру під рисунком «Рисунок А.1 – Назва рисунка».

Для тексту лістингу програми варто застосувати шрифт Courier New, розмір – 10 пт, колір – чорний, щільність тексту – звичайна. Номер лістингу відповідної функції та пояснювальний надпис розміщують послідовно, наприклад: Лістинг Б.1 (перший лістинг додатка Б). Назву лістинга розміщують по центру над кодом програми, крапка наприкінці не ставиться. На всі надані лістинги в тексті роботи повинні бути посилання або в дужках (лістинг Б.1), або по тексту.

Приклад оформлення лістингу:

Лістинг Б.1 – Програмний код основного модуля

```
#include <iostream>
#include <string.h>
using namespace std;

const int M = 3, N = 6;

void view_file(char* name) //Функція для перегляду файла
{
    char s[100];
    FILE* f;
    f=fopen(name, "rt");//Відкрити файл для читання як текстовий
```

```

if (f==NULL)
    { cout<<"Cannot open file to veiw\n"; return; }
cout << "\nПерегляд файла " << name << endl;
while (fgets(s, 100, f))
{
    //Зчитувати з файла послідовно рядки у s, допоки вони є у файлі
    s[strlen(s)-1]='\0';
    puts(s); // Вивести рядок на екран
}
fclose(f);
}

```

ПОРЯДОК ПІДГОТОВКИ ТА ЗАХИСТ КУРСОВОЇ РОБОТИ

До захисту курсової роботи допускаються студенти, які успішно, вчасно та в повному обсязі виконали усі пункти завдання та оформили пояснювальну записку до курсової роботи відповідно до вимог.

Курсова робота подається керівнику для перевірки у терміни, визначені у завданні на її виконання.

Захист курсової роботи відбувається на кафедрі відповідно затвердженого графіка. Для захисту курсової роботи студент готує доповідь для виступу та ілюстративний матеріал до неї.

Здобувач подає на розгляд:

- пояснювальну записку до курсової роботи;
- програмну реалізацію виконання завдання роботи відповідно до індивідуального варіанту.

Під час захисту студент повинен:

- чітко знати тему, мету та завдання роботи;
- пояснити суть використаних методів та засобів розв'язування поставленої задачі;
- надати і пояснити розроблений алгоритм;
- обґрунтувати використання програмних засобів;
- визначити коло можливих користувачів програмного проєкту;
- описати вхідну та вихідну інформацію для кожного завдання реалізованого у системі;
- пояснити ключові фрагменти діалогу програмної системи з користувачем та продемонструвати результати вирішення всіх поставлених завдань;
- сформулювати висновки виконаної курсової роботи.

Доповідь студента на захисті триває до 8–10 хвилин. Після доповіді студент дає вичерпні відповіді на питання, обґрунтовано та наполегливо відстоює свою точку зору.

Автор курсової роботи має продемонструвати вміння: логічно та аргументовано викладати матеріал; розробляти словесні та графічні алгоритми для розв'язання поставлених задач; коректно використовувати програмні засоби; проводити власні дослідження; володіти навичками узагальнення; формулювати висновки; працювати з інформаційними джерелами; ініціювати та обґрунтовувати інноваційні підходи та напрями розв'язання задачі, що досліджується.

При оцінюванні роботи буде враховуватись якість її виконання та оформлення курсової роботи відповідно до вимог, рівень складності розв'язуваних задач, правильність отриманих результатів, доповідь здобувача і повнота його відповідей на поставлені питання.

ПРИКЛАД ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Завдання

1. Створити за допомогою програми *Блокнот* текстовий файл, наприклад, з ім'ям *F1.txt*, та заповнити його числовими даними у вигляді матриці розмірності 3×5 . Написати мовою C програмний код для формування із даних текстового файлу числової матриці та вирішення таких завдань:

- створити вектор середньо арифметичних значень максимального і мінімального елементів стовпців;
- обчислити суму елементів останнього рядка матриці.

Програмно записати результати обчислень у новий файл, наприклад, з ім'ям *F2.txt*.

2. Створити за допомогою програми *Блокнот* новий текстовий файл, наприклад, з ім'ям *F3.txt*, та заповнити його текстовими даними.

Створити мовою C функцію для визначення перших слів кожного рядка у файлі *F3.txt*. Програмно вивести список цих слів на екран, як результат опрацювання текстових даних та записати у файл *F2.txt*.

Приклад опису функції визначення перших слів кожного рядка

first_word()

Функція `first_word()` визначає перші слова кожного рядка у файлі з ім'ям *F3.txt*, виводить їх на екран та у файл з ім'ям *F2.txt*.

Аргументами функції є: рядкова змінна `name3` – назва фізичного файлу з текстовими даними, створеного у Блокноті для цієї частини роботи; `name2` – ім'я файлу з результатами відбору[9].

Функція має тип `void`, тобто не повертає жодного значення засобами `return`.

Локальні змінні:

- `s` – рядок довжиною до 100 символів для зчитування тексту із файлу;
- `t` – рядок довжиною до 100 символів для визначення першого слова;
- `f2` та `f3` – файлові змінні, за допомогою яких здійснюється доступ до файлів.

Словесний алгоритм роботи функції:

1. Відкрити файли:

```
f3 = fopen(name3, "rt");
```

```
f2 = fopen(name2, "at");
```

Функція `fopen()` відкриває файл. Перший аргумент `name3` або `name2` – ім'я файлу, що відкривається; другий аргумент `"rt"` або `"at"` – це режим відкривання:

`"rt"` – файл відкривається як текстовий для зчитування даних;

`"at"` – файл відкривається як текстовий для дописування даних у кінець.

2. Перевірити, чи відкрилися файли. Якщо ні, то вивести на екран відповідне повідомлення і перервати виконання програми:

```
if (f3 == NULL)
{
    cout << "Cannot open file to veiw\n";
    return;
}
if (f2 == NULL)
{
    cout << "Cannot open file to append\n";
    return;
}
```

3. Вивести повідомлення про характер розв'язуваної задачі на екран та у файл із результатами:

```
cout << "\nПерші слова усіх рядків:\n";
fputs("\nПерші слова усіх рядків:\n", f2);
```

4. У циклі **while** зчитувати з файла послідовно рядки у *s*, допоки вони є у файлі (до кінця файла):

```
while (fgets(s, 100, f3))
```

Черговий рядок зчитується з файла за допомогою функції

```
fgets(s, 100, f3)
```

Вона має три аргументи:

s – рядкова змінна, в яку записується рядок, зчитаний із файла;

100 – максимальна кількість символів, які можуть бути зчитані за один раз;

f3 – файл, з якого зчитуються дані.

5. Виокремити перше слово з рядка *s* і записати його у рядок *t*. Розділювачі слів зазначено другим аргументом функції **strtok**:

```
t = strtok(s, " ,.;-^=:!?");
```

6. Результат вивести на екран та у файл з результатами відбору.

```
puts(t);
```

```
fprintf(f2, "%s\n", t);
```

Тут: *t* – рядок, який містить перше слово;

"%s\n" – формат виведення: "%s" використовується для позначення рядка,

"\n" – формує Enter, тобто слова виводитимуться у стовпчик;

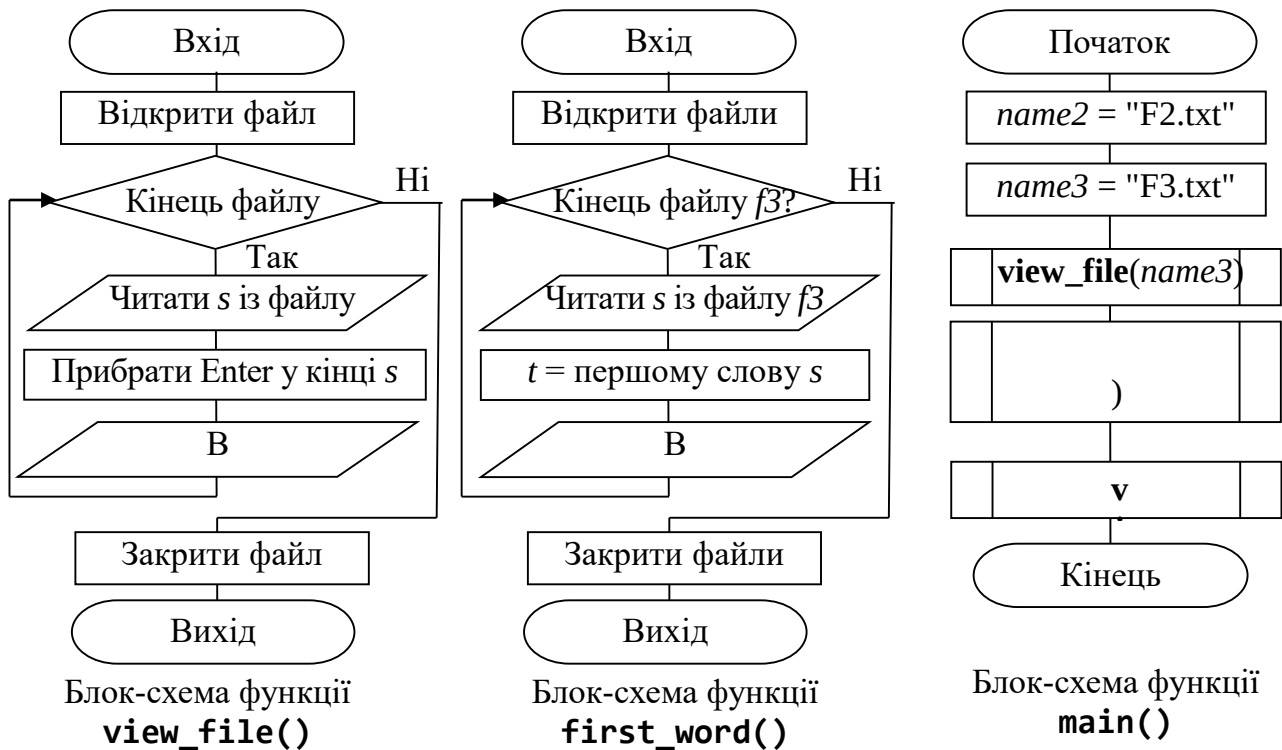
f2 – вказівник на файл з результатами.

7. Після закінчення роботи з файлами закрити їх.

```
fclose(f2);
```

```
fclose(f3);
```

Приклади схем алгоритмів роботи функцій



Програмний код у онлайн компіляторі C++ або у VS

```

#include <iostream>
#include <stdlib.h>
#include <string.h>
using namespace std;

const int M = 3, N = 5;
// Переглянути файл
void view_file(char* name)
{
    char s[100];    FILE* f;
    f = fopen(name, "rt"); // Відкрити файл для читання як текстовий
    if (f == NULL) { puts("Cannot open file to veiw"); return; }
    printf("\nПерегляд файла: %s\n", name);
    while (fgets(s, 100, f))
    { // Зчитувати з файла послідовно рядки у s, допоки вони є у файлі
        s[strlen(s) - 1] = '\0';
        puts(s); // Вивести рядок на екран
    }
    fclose(f);
}
  
```

```

// Заповнити матрицю значеннями із файла
void create_matrix(double matr[M][N], char* name)
{
    FILE* f;
    char s[100], * t;
    int i = 0, j;
    f = fopen(name, "rt"); // Відкрити файл для читання як текстовий
    if (f == NULL) { puts("Cannot open file to veiw"); return; }
    while (fgets(s, 100, f))
    {
        j = 0;
        t = strtok(s, " \t");
        while (t != NULL)
        {
            matr[i][j] = atof(t);
            t = strtok(NULL, " \t");
            j++;
        }
        i++;
    }
    fclose(f);
    return;
}

// Вивести матрицю на екран та у файл
void output_matrix(double matr[M][N], char* name2)
{
    int i, j;
    FILE* f2;
    f2 = fopen(name2, "wt"); // Створити файл як текстовий
    if (f2 == NULL) { puts("Cannot open file"); return; }
    printf("\n\nПерегляд матриці:\n");
    fprintf(f2, "Перегляд матриці:\n");
    for (i = 0; i < M; i++)
    {
        for (j = 0; j < N; j++)
        {
            // Вивести елемент матриці на екран
            printf("%7.1f\t", matr[i][j] );
            // Записати елемент у файл
            fprintf(f2, "%7.1f\t", matr[i][j]);
        }
        puts("");
        fprintf(f2, "%s", "\n");
    }
    fclose(f2);
    return;
}

```

// Створити вектор середнє арифметичних максимуму і мінімуму стовпців

```
void create_vector(double matr[M][N], double vekt[N], char* name2)
{
    FILE* f2;
    f2 = fopen(name2, "at"); // Відкрити файл для читання як текстовий
    if (f2 == NULL)
    {
        puts("Cannot open file to append data\n"); return;
    }
    puts("\nСтворений вектор:");
    fprintf(f2, "\nСтворений вектор:\n");
    double max, min;
    for (int j = 0; j < N; j++)
    {
        max = matr[0][j];
        min = matr[0][j];
        for (int i = 0; i < M; i++)
        {
            if (matr[i][j] > max) max = matr[i][j];
            if (matr[i][j] < min) min = matr[i][j];
        }
        vekt[j] = (max + min) / 2;
        printf("%7.1f\t", vekt[j]); // Вивести елемент вектора на екран
        fprintf(f2, "%7.1f\t", vekt[j]); // та у файл
    }
    fclose(f2);
}
```

// Сума елементів останнього рядка матриці

```
void sum(double matr[M][N], char* name2)
{
    FILE* f2;
    f2 = fopen(name2, "at"); // Відкрити файл для читання як текстовий
    if (f2 == NULL)
    {
        puts("Cannot open file to append data\n"); return;
    }
    double sum = 0;
    for (int j = 0; j < N; j++)
        sum += matr[M - 1][j];
    // Виведення результатів обчислень на екран та у файл
    printf("\n\nСума останнього рядка матриці = %7.1f\n", sum );
    fprintf(f2, "\n\nСума елементів останнього рядка = %7.1f\n", sum);
    fclose(f2);
}
```

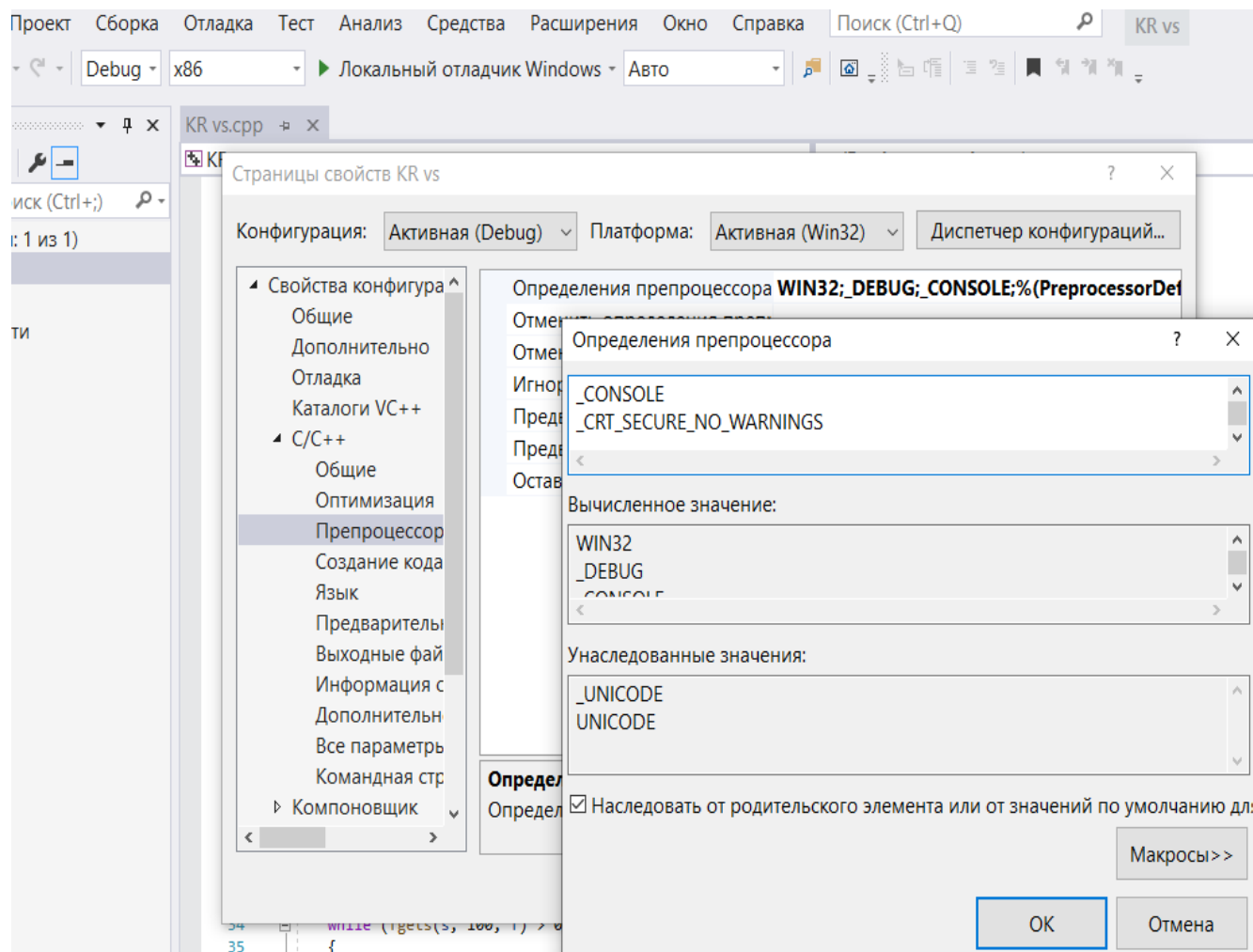
```

// Перше слово кожного рядка. Аргументи функції:
// name3 - ім'я файла з текстовими даними, створений у Блокноті для цієї
// частини роботи; name2 - ім'я файла з результатами відбору.
void first_word(char* name3, char* name2)
{
    FILE* f3, * f2; // Оголошення вказівників на файли F3 та F2
    f3 = fopen(name3, "rt"); // Відкрити текстовий файл для читання
    if (f3 == NULL) { puts("Cannot open file to view"); return; }
    f2 = fopen(name2, "at"); // Відкрити файл F3 для дописування
    if (f2 == NULL) { puts("Cannot open file to append\n"); return; }
    char s[100], * t = new char[100];
    puts( "\nПерші слова усіх рядків:");
    fputs("\nПерші слова усіх рядків:\n", f2);
    // Зчитувати з файла послідовно рядки у s, допоки вони є у файлі
    while (fgets(s, 100, f3))
    {
        t = strtok(s, " ,.;-^=:!?" ); // Виокремити перше слово,
        puts(t); // вивести його на екран
        fprintf(f2, "%s\n", t); //записати перше слово у файл
    }
    fclose(f2); // Закрити файл F2
    fclose(f3); // Закрити файл F3
}

int main()
{
    setlocale(0, ".1251");
    double A[M][N], B[N];
    // Оголошення імен файлів
    // Якщо файл розмістити у кореневій папці диска D,
    // то оголошення матиме вигляд: char name1[]="D:\\F1.txt";
    char name1[] = "F1.txt";
    char name2[] = "F2.txt";
    char name3[] = "F3.txt";
    view_file(name1); // Перегляд файла
    create_matrix(A, name1); // Заповнити матрицю числами із файла
    output_matrix(A, name2); // Вивести матрицю на екран та у файл
    // Створити вектор середнє арифметичних максимуму і мінімуму стовпців
    create_vector(A, B, name2);
    sum(A, name2); // Сума елементів останнього рядка
    view_file(name3); // Переглянути файл F3
    first_word(name3, name2); // Перше слово кожного рядка
    view_file(name2); // Переглянути файл F2
    return 0;
}

```

У разі виникнення помилок компіляції у VS для усунення попереджень про нерекомендовані функції варто виконати таке. У меню *Проект* вибрати команду *Властивості* (або натиснути клавіші Alt + F7). У вікні *Сторінки властивостей*, що відкриється, вибрати *C/C++* → *Препроцесор*, де у полі *Визначення препроцесора* додати значення `_CRT_NONSTDC_NO_WARNINGS`. У версії з англomовним інтерфейсом послідовність команд така: *Configuration Properties* → *C/C++* → *Preprocessor* → *Preprocessor Definitions* → `_CRT_NONSTDC_NO_WARNINGS`.



Результати виконання програми

```

    Реве та стогне Дніпр широкий,
    Сердитий вітер завива,
    Додолу верби гне високі,
    Горами хвилю підійма.
    Ще треті півні не співали,
    Ніхто нігде не гомонів,
    Сичі в гаю перекликались,
    Та ясен раз у раз скрипів.

Перші слова усіх рядків:
    Реве
    Сердитий
    Додолу
    Горами
    Ще
    Ніхто
    Сичі
    Та

Перегляд файлу F2.txt
Перегляд матриці:
      5.0      8.0     -2.0      1.0      6.0
     -12.0     -1.0      0.0      0.0      1.0
      2.0      9.0     -1.0      0.0      2.0

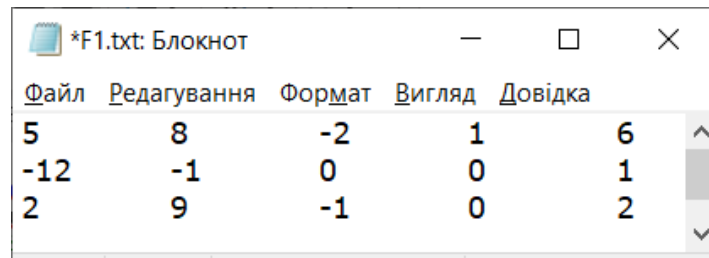
Створений вектор:
     -3.5      4.0     -1.0      0.5      3.5

Сума елементів останнього рядка =      12.0

Перші слова усіх рядків:
    Реве
    Сердитий
    Додолу
    Горами
    Ще
    Ніхто
    Сичі
    Та

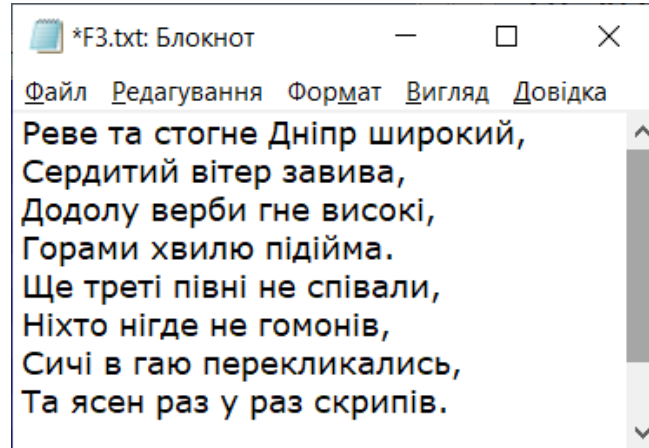
```

Рисунок 1 – Скріншот консольного вікна з результатами роботи проєкту



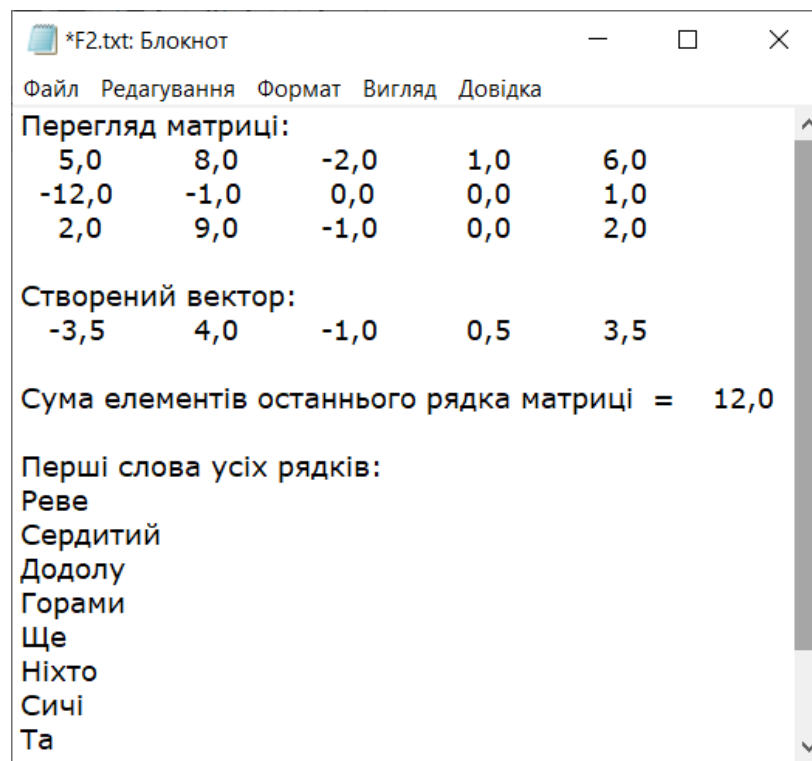
5	8	-2	1	6
-12	-1	0	0	1
2	9	-1	0	2

Рисунок 2 – Вигляд текстового файла з вхідною матрицею числових елементів



Рече та стогне Дніпр широкий,
Сердитий вітер завива,
Додолу верби гне високі,
Горами хвилю підійма.
Ще треті півні не співали,
Ніхто нігде не гомонів,
Сичі в гаю перекликались,
Та ясен раз у раз скрипів.

Рисунок 3 – Вигляд вхідного текстового файла з текстовими даними для подальшого програмного опрацювання



Перегляд матриці:

5,0	8,0	-2,0	1,0	6,0
-12,0	-1,0	0,0	0,0	1,0
2,0	9,0	-1,0	0,0	2,0

Створений вектор:

-3,5	4,0	-1,0	0,5	3,5
------	-----	------	-----	-----

Сума елементів останнього рядка матриці = 12,0

Перші слова усіх рядків:

Рече
Сердитий
Додолу
Горами
Ще
Ніхто
Сичі
Та

Рисунок 4 – Вигляд звітнього текстового файла з результатами програмного опрацювання

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Трофименко О.Г., Прокоп Ю.В., Задерейко О.В. Алгоритмізація та програмування : навч.-метод. посіб. Одеса: Фенікс, 2020. 310 с. URL: <http://dspace.onua.edu.ua/handle/11300/12345>.
2. Васильєв О. Програмування на С++ в прикладах і задачах: Навч. посіб. О.Васильєв. - Київ :Видавництво Ліра-К, 2023. -382с.
3. Буката Л.М. Основи програмування : Роб. прогр. навч. дисципл. [ОПП «Інженерія програмного забезпечення» зі спеціальності «Інженерія програмного забезпечення» для здобувачів першого (бакалаврського) рівня вищої освіти] / Розробн. : Л.М. Буката. Одеса : ДУІТЗ, 2023. 36 с.
4. С++. Основи програмування. Теорія та практика: підручник / О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката [та ін.] за ред. О. Г. Трофименко. Одеса : Фенікс, 2010. 544 с.
5. С++. Теорія та практика: навч. посіб. з грифом МОНУ / О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката [та ін.] за ред. О. Г. Трофименко. Одеса : ВЦ ОНАЗ, 2011. 587 с.
6. Трофименко О. Г., Прокоп Ю. В., Буката Л. М. Основи програмування. Базові алгоритми : метод. вказівки для лаб. і практ. робіт. Ч. 1. Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2014. 108 с.
7. Трофименко О. Г., Прокоп Ю.В., Буката Л.М. Основи програмування. Програмування структурованих даних : метод. вказівки для лаб. і практ. робіт. Ч. 2. Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2014. 130 с.
8. Трофименко О.Г., Прокоп Ю.В., Буката Л.М. Основи програмування. Програмне опрацювання файлів : метод. вказівки для лаб. і практ. робіт. Ч. 3. Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2016. 80 с.
9. Трофименко О.Г., Прокоп Ю.В., Буката Л.М. Комп'ютерні технології та програмування. Програмне опрацювання файлів : метод. вказівки для виконання курсової роботи. Одеса: ДУІТЗ, 2021. 38 с.
10. ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання». Чинний від 2017-07-01. К.: ДП «УкрНДНЦ». 2016. – 31 с.
11. ДСТУ ISO/IEC 2382-5:2005 Інформаційні технології. Словник термінів. Частина 5. Подання даних (ISO/IEC 2382-5:1999, IDT).
12. ДСТУ ISO/IEC 2382-15:2005 Інформаційні технології. Словник термінів. Частина 15. Мови програмування (ISO/IEC 2382-15:1999, IDT).
13. ДСТУ 2873-94 Системи оброблення інформації. Програмування. Терміни та визначення.
14. ДСТУ 4302:2004 Інформаційні технології. Настанови щодо документування комп'ютерних програм (ISO/IEC 6592:2000 MOD).

ДОДАТОК А

Приклад оформлення титульної сторінки курсової роботи

Міністерство освіти і науки України
Державний університет інтелектуальних технологій і зв'язку
Кафедра інженерії програмного забезпечення

КУРСОВА РОБОТА
з дисципліни «Основи програмування»
за темою: «Створення багатомодульних програмних проєктів
для опрацювання даних у файлах»

студента 1-го курсу, групи ПЗ-1.01
спеціальності "Інженерія програмного за-
безпечення"

Іванюка Богдана Миколайовича

Керівник доцент, к.т.н. Шевченко М.Г.

Національна шкала _____

Кількість балів: _____ Оцінка ECTS _____

(підпис) Шевченко М. Г.
(прізвище та ініціали)

Одеса – 2024 р.

ДОДАТОК Б

Приклад завдання до курсової роботи

ЗАВДАННЯ ДО КУРСОВОЇ РОБОТИ

з дисципліни «Основи програмування»

студента Іванюка Богдана Миколайовича

спеціальності F2 «Інженерія програмного забезпечення»

першого курсу групи ІПЗ-1.01.

Тема «Створення багатомодульних програмних проектів для опрацювання даних у файлах»

Варіант 16

Вхідні дані:

- текстовий файл *F1.txt* із числовими даними у вигляді матриці розмірності M на N (M – кількість рядків, N – кількість стовпців);
- файл *F2.txt* з текстом (не менше 10 рядків).

Курсова робота виконується в такому обсязі:

I. Розрахунково-пояснювальна записка:

- аналіз засобів та методів розробки;
- словесний опис алгоритмів основного модуля;
- опис функцій розробленої бібліотеки MyLibrary;
- аналіз досягнутих результатів;
- інструкція з використання програми.

II. Графічна частина:

- схеми алгоритмів основного модуля для виконання завдань;
- схеми алгоритмів функцій бібліотеки MyLibrary.

Індивідуальне завдання:

1. Розробити алгоритми та написати програмний код для формування із даних текстового файла *F1.txt* числової матриці та вирішення таких завдань:

- створити вектор середнє арифметичних значень максимуму і мінімуму стовпців;
- обчислити суму елементів останнього рядка матриці.

2. Програмно записати результати обчислень у новий файл, наприклад, з ім'ям *F2.txt*.

3. Розробити алгоритми та написати програмний код для визначення перших слів кожного рядка у файлі *F3.txt*. Програмно вивести список цих слів на екран, як результат опрацювання текстових даних та записати у файл *F2.txt*.

КАЛЕНДАРНИЙ ПЛАН

№	Завдання	Термін виконання
1	Ознайомлення з загальним та індивідуальним завданням до курсової роботи. Підготування вхідних файлів	19.10.2024- 25.10.2024
2	Частина 1. Опрацювання матриць	26.10.2024- 12.11.2024
3	Частина 2. Опрацювання динамічних масивів	13.11.2024- 25.11.2024
4	Частина 3. Опрацювання рядкових даних	26.11.2024- 08.12.2024
5	Частина 4. Опрацювання текстових файлів	09.12.2024- 19.12.2024
5	Оформлення пояснювальної записки	20.12.2024- 25.12.2024

Навчально-методичне видання

Буката Людмила Миколаївна

**Створення багатомодульних програмних проєктів
для опрацювання даних у файлах**

Методичні рекомендації для виконання курсової роботи
з дисципліни

"Основи програмування"

для підготовки здобувачів першого (бакалаврського) рівня вищої освіти
у галузі знань «Інформаційні технології»
спеціальності Інженерія програмного забезпечення

Електронне видання

ДУІТЗ
65023, м. Одеса, вул Кузнечна, 1
Обсяг 2.38 др.арк., електронна версія