

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНИКА



МАЗУРЕНКО В.В., ДМИТРИШИН М.І., ВАСИЛИШИН П.Б.

СТРУКТУРНО-ПРОЦЕДУРНЕ ПРОГРАМУВАННЯ з PYTHON

ЛАБОРАТОРНИЙ ПРАКТИКУМ

УДК 004.424
ББК 32.973-018.1
М13

Рекомендовано Вченою радою факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника (протокол № 10 від 21.11.2023 р.).

Рецензенти:

Олександр МАХНЕЙ, кандидат фізико-математичних наук, доцент (Прикарпатський національний університет імені Василя Стефаника, Івано-Франківськ);

Роман ДМИТРИШИН, доктор фізико-математичних наук, професор (Прикарпатський національний університет імені Василя Стефаника, Івано-Франківськ).

М13 Мазуренко В.В., Дмитришин М.І., Васишин П.Б. Структурно–процедурне програмування з Python: Лабораторний практикум. – Ів.-Фр.: Голіней, 2023. – 76 с.

У лабораторному практикумі наведено індивідуальні завдання і методичні рекомендації до виконання лабораторних робіт з основ програмування мовою Python.

Для студентів спеціальності «прикладна математика». Практикум буде корисним також для студентів галузей знань «математика і статистика» та «інформаційні технології».

ЗМІСТ

ПЕРЕДМОВА	4
Лабораторна робота 1. ЛІНІЙНІ ПРОГРАМИ	5
Лабораторна робота 2. РОЗГАЛУЖЕНІ ПРОГРАМИ	10
Лабораторна робота 3. ЦИКЛІЧНІ ПРОГРАМИ	16
Лабораторна робота 4. СПИСКИ І КОРТЕЖІ	27
Лабораторна робота 5. СИМВОЛИ І РЯДКИ	38
Лабораторна робота 6. СЛОВНИКИ І МНОЖИНИ	46
Лабораторна робота 7. ОБРОБКА ВИНЯТКІВ	58
Лабораторна робота 8. СТВОРЕННЯ ФУНКЦІЙ	66
Лабораторна робота 9. ТЕКСТОВІ І БІНАРНІ ФАЙЛИ	72
СПИСОК ЛІТЕРАТУРИ	76

ПЕРЕДМОВА

Програмування як процес написання (coding), відлагодження (debugging) і тестування (testing) комп'ютерних програм для автоматизації (на основі запропонованих алгоритмів) завдань у різних предметних областях, допускає різні підходи до його реалізації. Тривалий час домінуючим підходом у програмуванні була парадигма *структурно-процедурного програмування*, відповідно до якої комп'ютерна програма має вигляд ієрархічної структури блоків з потоком керування (control flow), що реалізується за допомогою таких керуючих конструкцій, як послідовне виконання, галуження, цикл, а також виклик підпрограм та/або функцій. Структуровані програми легше створювати, відстежувати логіку їх виконання і відлагоджувати; скорочується термін їх розробки; є можливість повторного використання одного і того ж коду з різних місць програми; зазвичай, є швидшими через низький рівень абстракції.

Лабораторний практикум зі структурно-процедурного програмування є важливим етапом у навчанні здобувачів освіти основам програмування та формуванні у них алгоритмічного мислення. Він зорієнтований на використання мови програмування Python. Це високорівнева, інтерпретована мова програмування, яка відзначається простотою синтаксису і читабельністю коду. Її універсальність у поєднанні з численними бібліотеками робить Python ідеальним вибором для широкого спектру завдань — від автоматизації задач і наукових обчислень, аж до інтелектуального аналізу даних і веб-розробки. Водночас варто зауважити, що завдання лабораторного практикуму є інваріантними стосовно мови програмування і за потреби можуть бути використані для вивчення інших мов.

Практикум охоплює широкий спектр тем, починаючи від лінійних програм та закінчуючи роботою з файлами. Особлива увага приділяється керуючим конструкціям, що дозволяє здобувачам освіти освоїти логіку виконання програми через галуження і цикли. Практичне використання списків, кортежів, рядків, словників і множин розширює їхні можливості у роботі з різними типами і структурами даних. Окрім того, важливий аспект практикуму — це вивчення обробки виняткових ситуацій, що забезпечує студентам розуміння важливості відлагодження програм. Навики створення функцій та використання файлів і модулів сприяють розробці більш структурованих, читабельних і гнучких у плані організації коду програм. Для кожної лабораторної роботи вказана тема, короткий перелік необхідних для виконання роботи теоретичних знань, 15 варіантів індивідуальних або групових (для командної роботи) завдань та вимоги до оформлення звіту.

I. МЕТА ЛАБОРАТОРНОЇ РОБОТИ

формування навичок написання програм мовою Python для реалізації лінійних алгоритмів

II. Для виконання лабораторної роботи НЕОБХІДНО ЗНАТИ:

- синтаксис мови Python
- що таке ідентифікатор, константа (літерал та іменована константа), змінна, ключове слово, коментар до коду
- основні найпростіші інструкції: тотожна, присвоєння, введення, виведення, оператор %
- числові типи констант і змінних (int, float, complex) та способи їх визначення і перетворення
- арифметичні операції та їх пріоритет
- базові функції для числових типів та функції бібліотек math і cmath

РЕКОМЕНДОВАНА ЛІТЕРАТУРА: [1, п. 1.1-6], [2, п. 3.1-6], [3, § 1], [4, п. 2.7-10, 3.1-4], [5]

ДОДАТКОВА ЛІТЕРАТУРА: [6–9]

III. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Написати лінійні програми¹⁾, вважаючи, що вхідні дані вводяться користувачем коректно. Варіанти завдань наведені у таблиці, *номер варіанту є порядковим номером студента в електронному журналі академгрупи.*

Таблиця 1 Варіанти завдань

Варіант	Задачі	Варіант	Задачі
1	1 8 12	9	7 17 3
2	4 11 18	10	10 8 12
3	7 5 15	11	13 5 6
4	10 2 3	12	16 11 18
5	13 14 6	13	1 17 15
6	16 17 9	14	4 2 9
7	1 2 15	15	7 14 3
8	4 14 9	16	10 8 12

¹⁾ Використовувати галуження і цикли, а також методи для списків, кортежів, рядків, словників, множин заборонено.

I. МЕТА ЛАБОРАТОРНОЇ РОБОТИ

формування навичок написання програм мовою Python для реалізації розгалужених алгоритмів

II. Для виконання лабораторної роботи НЕОБХІДНО ЗНАТИ

- що таке потік керування у парадигмі структурного програмування
- логічні операції та операції відношення, їх пріоритет
- що таке логічний тип даних
- логіку умовного оператора if..else
- логіку каскадного галуження if..elif..else
- логіку тернарного умовного оператора ..if..else..

РЕКОМЕНДОВАНА ЛІТЕРАТУРА: [1, п. 1.7, 2.1], [2, п. 5.1-2], [3, § 2], [4, п. 6.1-2], [5]

ДОДАТКОВА ЛІТЕРАТУРА: [6–9]

III. Завдання на лабораторну роботу

Написати розгалужені програми¹⁾, вважаючи, що вхідні дані вводяться користувачем коректно. Варіанти завдань наведені у таблиці, *номер варіанту є порядковим номером студента в електронному журналі академгрупи.*

Таблиця 1 Варіанти завдань

Варіант	Задачі	Варіант	Задачі
1	1(a) 14 3	9	1(i) 11 3
2	1(b) 17 6	10	1(j) 13 6
3	1(c) 19 7	11	1(k) 14 7
4	1(d) 2 9	12	1(l) 17 9
5	1(e) 4 12	13	1(m) 19 12
6	1(f) 5 15	14	1(n) 4 15
7	1(g) 8 16	15	1(o) 8 16
8	1(h) 10 18	16	1(a) 11 18

¹⁾ Використовувати цикли, а також методи для списків, кортежів, рядків, словників, множин заборонено.

I. МЕТА ЛАБОРАТОРНОЇ РОБОТИ

формування навичок написання програм мовою Python для реалізації циклічних алгоритмів

II. Для виконання лабораторної роботи НЕОБХІДНО ЗНАТИ

- логіку циклу з передумовою `while`
- логіку циклу по колекції `for..in..`
- як побудувати колекцію цілочислових значень функцією `range()`
- команди переривання окремої ітерації в циклі і циклу загалом
- що таке рекурентні співвідношення першого і вищих порядків
- що таке системи рекурентних співвідношень
- методи наближених обчислень границь і коренів алгебричних рівнянь

РЕКОМЕНДОВАНА ЛІТЕРАТУРА: [1, п. 2.2-3], [2, п. 6.1-3], [3, § 3], [4, п. 6.3], [5]

ДОДАТКОВА ЛІТЕРАТУРА: [6–9]

III. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Написати циклічні програми¹⁾, вважаючи, що вхідні дані вводяться користувачем коректно. Варіанти завдань наведені у таблиці, **номер варіанту є порядковим номером студента в електронному журналі академгрупи.**

Таблиця 1 Варіанти завдань

Варіант	Задачі					Варіант	Задачі				
1	1	2	3	4	5	9	41	42	43	44	45
2	6	7	8	9	10	10	46	47	48	49	50
3	11	12	13	14	15	11	51	52	53	54	55
4	16	17	18	19	20	12	56	57	58	59	60
5	21	22	23	24	25	13	1	12	18	24	30
6	26	27	28	29	30	14	6	22	28	34	40
7	31	32	33	34	35	15	11	32	38	44	50
8	36	37	38	39	40	16	21	42	48	54	60

¹⁾ Використовувати методи для списків, кортежів, рядків, словників, множин заборонено.

I. МЕТА ЛАБОРАТОРНОЇ РОБОТИ

формування навичок роботи зі списками і кортежами мовою Python

II. Для виконання лабораторної роботи необхідно знати

- що таке список (list) і способи його генерування
- способи доступу до списку з допомогою індексації і зрізу
- операції над списками: конкатенація і зчеплення, перевірка входження елемента у список, аналіз і модифікація списку
- особливості роботи зі списком як змінюваною (mutable) структурою даних
- способи обходу списку: за індексами, за елементами, за парами індекс–елемент (enumerate)
- можливості моделювання матриць з допомогою списків і роботи з ними
- що таке кортеж (tuple) і чим він відрізняється від списку (list)
- способи генерування кортежів і операції над ними

РЕКОМЕНДОВАНА ЛІТЕРАТУРА: [1, п. 4.1-6], [2, п. 8.1-9], [3, § 4], [4, п. 4.1-16], [5]

ДОДАТКОВА ЛІТЕРАТУРА: [6–9]

III. Завдання на лабораторну роботу

Написати програми¹⁾, вважаючи, що вхідні дані вводяться користувачем коректно.
Номер варіанту є порядковим номером студента в електронному журналі академгрупи.

Варіант 1.

1. Для введених з клавіатури натурального числа N та дійсних чисел b і q ($\neq 0, 1$) згенерувати і вивести на екран список N членів геометричної прогресії з першим членом b і знаменником q . Обчислити їх суму (без використання функції `sum()`).

ТЕСТ ВХІД:

7 ↵ 0.5 ↵ -2 ↵

ВИХІД:

[0.5, -1.0, 2.0, -4.0, 8.0, -16.0, 32.0]

21.5

2. Задані N точок на площині. Вивести на екран дві точки, відстань між якими найбільша, та саму відстань.

ТЕСТ ВХІД:

3 ↵ # кількість точок

0 ↵ 0 ↵

3 ↵ 0 ↵

0 ↵ 4 ↵

ВИХІД:

(3, 0)

(0, 4)

5

¹⁾ Використовувати методи для рядків, словників, множин заборонено.

I. МЕТА ЛАБОРАТОРНОЇ РОБОТИ

формування навичок роботи із символами і рядками мовою Python

II. Для виконання лабораторної роботи НЕОБХІДНО ЗНАТИ

- що таке таблиця кодування символів і які кодування використовуються у мові Python
- для чого призначені escape-послідовності, найчастіше вживані екрановані символи та їх значення
- основні операції із символами
- що таке рядок (str) і способи його генерування
- операції над рядками: успадковані від кортежів, конкатенація і зчеплення, перевірка входження символів, пошук і заміна, аналіз і модифікація, розбиття і склейка
- що таке лексикографічне впорядкування рядків
- методи форматування рядків

РЕКОМЕНДОВАНА ЛІТЕРАТУРА: [1, п. 5.3], [2, п. 3.4, 2.1-8], [3, § 5], [4, п. 3.3], [5]

ДОДАТКОВА ЛІТЕРАТУРА: [6–9]

III. Завдання на лабораторну роботу

Написати програми¹⁾, вважаючи, що вхідні дані вводяться користувачем коректно.
Номер варіанту є порядковим номером студента в електронному журналі академгрупи.

Варіант 1.

1. Визначити кількість різних символів у введеному з клавіатури рядку і вивести на екран ці символи через пропуск.

ТЕСТ ВХІД:

AbraKadabra ↵

ВИХІД:

5

A b r k d

2. Введений арифметичний вираз складається із чисел, знаків + і – (відокремлених від чисел пропуском) та круглих дужок.

(a) вивести на екран значення цього арифметичного виразу;

(b) підрахувати кількість доданків у виразі.

ТЕСТ ВХІД:

346 – (127 + 85) – 12 ↵

ВИХІД:

(a) 232

(b) 4

¹⁾ Використовувати методи для словників і множин заборонено.

I. МЕТА ЛАБОРАТОРНОЇ РОБОТИ

формування навичок роботи зі словниками і множинами мовою Python

II. Для виконання лабораторної роботи необхідно знати

- що таке словник (dict) і способи його генерування, до якого типу (змінюваний чи незмінюваний) він належить
- як здійснюється доступ до елементів словника
- операції над словниками: перевірка приналежності ключа або значення до словника, вибір елементів чи колекцій словника, аналіз і модифікація словника
- способи обходу словника з допомогою: ключа (key), значення (value), запису (item)
- що таке множина (set) і способи її генерування
- до якого типу (змінюваний чи незмінюваний) належить множина і чим вона відрізняється від словника
- основну відмінність між такими колекціями як словник, множина та список, кортеж, рядок
- операції над множинами: відношення, перетин, об'єднання, різниця, додавання і видалення елементів
- що таке незмінювані множини (frozenset) і чим вони відрізняються від змінюваних множин (set)

РЕКОМЕНДОВАНА ЛІТЕРАТУРА: [1, п. 5.1-2], [2, п. 8.1-9], [3, § 6], [4, п. 5.1-11], [5]

ДОДАТКОВА ЛІТЕРАТУРА: [6–9]

III. Завдання на лабораторну роботу

Написати програми, вважаючи, що вхідні дані вводяться користувачем коректно.
Номер варіанту є порядковим номером студента в електронному журналі академгрупи.

Варіант 1.

1. Задано словник студентів групи, у якому ключем є порядковий номер студента, а значенням список мов, які знає студент. Вивести на екран мову(-и), яку(-і) знають усі студенти групи.

ТЕСТ ВХІД:

```
{ 1 : [Ukrainian, English, Python],  
 2 : [English, Ukrainian, Python, German],  
 3 : [French, Python, Ukrainian, Polish, Spanish],  
 4 : [Python, Ukrainian, Spanish],  
 5 : [Ukrainian, Python]}
```

ВИХІД:

Ukrainian, Python

I. МЕТА ЛАБОРАТОРНОЇ РОБОТИ

формування навичок роботи з помилками і винятковими ситуаціями мовою Python

II. Для виконання лабораторної роботи необхідно знати

- які типи помилок можуть виникати у програмах
- що генерується в результаті виникнення виняткової ситуації
- як відбувається обробка винятків з допомогою оператора `try..except..else..finally`
- ієрархію стандартних класів винятків
- як можна створити екземпляр винятків того чи іншого типу
- синтаксис перевірки твердження `assert` та ініціювання власного винятку `raise`
- що таке менеджер контексту і як його реалізувати з допомогою оператора `with..as`

РЕКОМЕНДОВАНА ЛІТЕРАТУРА: [1, п. 2.4], [2, п. 10.11-12], [3, § 7], [4, п. 6.7], [5]

ДОДАТКОВА ЛІТЕРАТУРА: [6–9]

III. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Написати програми *з обробкою потенційних виняткових ситуацій* (при введенні вхідних даних користувачем або впродовж життєвого циклу програми). *Номер варіанту є порядковим номером студента в електронному журналі академ-групи.*

Варіант 1.

1. Із заданою точністю $\varepsilon > 0$ обчислити мажорне золоте число Φ ¹⁾, використовуючи границю $\Phi = \lim_{n \rightarrow \infty} \frac{F_n}{F_{n-1}}$, де F_n — n -е число Фібоначчі, що задовольняє відповідне рекурентне співвідношення.

ТЕСТ-1 ВХІД:

-1e-6 ↵

ТЕСТ-2 ВХІД:

1e-6 ↵

ВИХІД:

Некоректні вхідні дані!

ВИХІД:

1.6180342

2. Для введених з клавіатури натурального числа N та дійсних чисел b і q ($\neq 0, 1$) згенерувати і вивести на екран список N членів геометричної прогресії з першим членом b і знаменником q . Обчислити їх суму.

¹⁾ Золотим поділом відрізка довжини c ($= a + b$) на дві нерівні частини з довжинами a та b ($a > b$) називають поділ, при якому відношення цілого відрізка до його більшої частини, дорівнює відношенню більшої частини до його меншої частини, тобто $\frac{a+b}{a} = \frac{a}{b}$. Відношення $\Phi = \frac{a}{b} = \frac{\sqrt{5}+1}{2}$ та $\varphi = \frac{1}{\Phi} = \frac{\sqrt{5}-1}{2}$, для яких $\Phi - \varphi = 1$, називають золотими числами (мажорним і мінорним відповідно).

Лабораторна робота 8

СТВОРЕННЯ ФУНКЦІЙ

I. МЕТА ЛАБОРАТОРНОЇ РОБОТИ

формування навичок створення і використання функцій мовою Python

II. Для виконання лабораторної роботи НЕОБХІДНО ЗНАТИ

- що таке підпрограми (процедури і функції), основні етапи опису функцій
- як повернути кілька значень однією функцією
- що таке формальні і фактичні аргументи функції і як відбувається їх передача у функцію
- що таке позиційне і ключове співставлення формальних і фактичних аргументів функції
- як задавати типові (за замовчуванням) значення аргументів
- як описуються функції з довільною кількістю аргументів
- яка різниця між локальними і глобальними змінними у тілі функції і які області їх видимості
- що таке анонімні (λ -) і рекурсивні функції та функції-генератори
- призначення і опис декораторів для функцій

РЕКОМЕНДОВАНА ЛІТЕРАТУРА: [1, п. 3.1-9], [2, п. 10.1-10], [3, § 8], [4, п. 6.6], [5]

ДОДАТКОВА ЛІТЕРАТУРА: [6–9]

III. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Написати програми з обробкою потенційних виняткових ситуацій для виконання наступних завдань.. *Номер варіанту є порядковим номером студента в електронному журналі академгрупи.*

Варіант 1.

1. Два натуральні числа називають "близнюками", якщо вони прості і відрізняються одне від одного на 2 (числа 41 і 43 є близнюками). Вивести на екран усі пари близнюків із відрізка $[n, 2n]$, де n — введене натуральне число. У програмі описати і використати функцію $\text{IsPrime}(k)$, яка визначає, чи є число k простим.
2. Введені перший член і різниця арифметичної прогресії. Описати і застосувати рекурсивну функцію для обчислення n -го члена цієї прогресії. Використовуючи декоратор, вивести на екран відповідь у вигляді: *n -ий член заданої арифметичної прогресії є <тут повернути результат виклику рекурсивної функції>.*
3. Описати і застосувати (анонімну) lambda-функцію, яка для заданого рядка s повертає його довжину.

I. МЕТА ЛАБОРАТОРНОЇ РОБОТИ

формування навичок роботи з файлами мовою Python

II. Для виконання лабораторної роботи НЕОБХІДНО ЗНАТИ

- що таке файл і його ознаки
- три основні етапи роботи з файлами: відкриття, читання/запис, закриття
- інструкції для відкриття і закриття файлу та режими роботи з файлом
- основні операції з файлами
- особливості роботи з текстовими і бінарними файлами
- основні операції з файловою системою

РЕКОМЕНДОВАНА ЛІТЕРАТУРА: [2, п. 7.1-4], [3, § 9], [4, п. 8.1-5], [5]

ДОДАТКОВА ЛІТЕРАТУРА: [6–9]

III. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Працюючи у команді (номер варіанту виконує трійка студентів відповідно до таблиці нижче, перший зі студентів є TeamLeader, що розподіляє і контролює роботу, захист роботи від команди), написати програми з обробкою потенційних виняткових ситуацій для виконання наступних завдань.

Таблиця 1 Варіанти завдань

Варіант	Номер студента		
1	1	6	11
2	7	12	2
3	13	3	8
4	4	9	14
5	10	15	5

Варіант 1.

1. У вхідному файлі `data1.txt` міститься послідовність цілих чисел, які розділені пробілами. Дозаписати у цей файл такі рядки:
 - (a) від'ємні числа з послідовності;
 - (b) кількість нулів;
 - (c) додатні числа з послідовності;
 - (d) шлях до каталогу, у якому міститься файл, час створення і розмір файлу.

2. У вхідному файлі `data2.txt` містяться координати точок на площині. Дозаписати у цей файл рядки з інформацією про найвіддаленішу і найближчу пару точок серед заданих.

Приклад файлу `data2.txt`

(-5, 2)

(1, -4)

(0, 0)

(-3, -7)

3. Задано словник студентів групи, у якому ключем є порядковий номер студента, а значенням список мов, які знає студент. Виконати такі завдання:

(а) серіалізувати цей словник у бінарний файл `binary.data`;

(б) десеріалізувати дані з файлу і вивести на екран мову(-и), яку(-і) знають усі студенти групи.

Варіант 2.

1. У вхідному файлі `data1.txt` міститься послідовність цілих чисел, які розділені пропусками. Дозаписати у цей файл такі рядки:

(а) числа, що є менші за їх середнє арифметичне;

(б) кількість чисел, що дорівнюють середньому арифметичному;

(с) числа, що є більші за середнє арифметичне;

(д) шлях до каталогу, у якому міститься файл, час створення і розмір файлу.

2. У вхідному файлі `data2.txt` містяться результати кожної з 4-х чвертей баскетбольних матчів. Дозаписати у цей файл рядки з інформацією про загальний рахунок матчу і різницю очок, набраних командами.

Приклад файлу `data2.txt`

26:17 13:15 19:11 14:16

15:16 18:17 10:12 14:12

20:21 24:22 18:15 16:25

3. Задано словник з N англійських слів з можливими повторами слів. Виконати такі завдання:

(а) серіалізувати цей словник у бінарний файл `input.data`;

(б) десеріалізувати дані з файлу і вивести на екран словник без повторів слів, впорядкованих лексикографічно, і кількість слів у ньому.

Варіант 3.

1. У вхідному файлі `data1.txt` міститься послідовність цілих чисел, які розділені пропусками. Дозаписати у цей файл такі рядки:

(а) кількість членів послідовності, модулі яких є простими числами : вивести такі члени послідовності;

(б) кількість членів послідовності, модулі яких є складеними числами : вивести такі члени послідовності;

(с) кількість членів послідовності, модулі яких не є ні простими, ні складеними числами : вивести такі члени послідовності.

(д) шлях до каталогу, у якому міститься файл, час створення і розмір файлу.

2. У вхідному файлі `data2.txt` міститься перелік наявних у шкільній бібліотеці книг. Дозаписати у цей файл рядки з інформацією про книги в одному примірнику та кількість таких книг у списку.

Приклад файлу `data2.txt`

Тарас Шевченко : “Кобзар” — 1 прим.

Юрій Винничук : “Легенди Львова” — 3 прим.

Ліна Костенко : “Гіацинтове сонце” — 2 прим.

3. Задано словник міст, у якому ключем є країна, а значенням список окремих міст цієї країни. Виконати такі завдання:
 - (а) серіалізувати цей словник у бінарний файл `binary.data`;
 - (б) десеріалізувати дані з файлу і для введеного з клавіатури міста, вивести на екран країну, де воно знаходиться.

Варіант 4.

1. У вхідному файлі `data1.txt` міститься непорожня послідовність цифр (0 – 11) дванадцяткової системи числення, які розділені пропусками. Дозаписати у цей файл такі рядки:
 - (а) вивести a разів цифру a , де a — перша цифра з послідовності;
 - (б) вивести b разів цифру b , де b — друга цифра з послідовності;
 - (с) і т.д. аж до останньої цифри з послідовності;
 - (д) шлях до каталогу, у якому міститься файл, час створення і розмір файлу.
2. У вхідному файлі `data2.txt` міститься перелік телефонних розмов особи з абонентами різних операторів. Дозаписати у цей файл рядки з інформацією про розмови з абонентами від вибраного оператора (введеного з клавіатури) і середню тривалість розмов.
Приклад файлу `data2.txt`
1.12.2021 — 58 с (Vodafone)
1.12.2021 — 81 с (Kyivstar)
1.12.2018 — 36 с (Lifecell)
3. Задано словник студентів групи, у якому ключем є порядковий номер студента, а значенням список мов, які знає студент. Виконати такі завдання:
 - (а) серіалізувати цей словник у бінарний файл `binary.data`;
 - (б) десеріалізувати дані з файлу і вивести на екран усі мови, які знають студенти групи.

Варіант 5.

1. У вхідному файлі `data1.txt` міститься непорожня послідовність цілих чисел, які розділені пропусками. Дозаписати у цей файл такі рядки:
 - (а) кількість чисел з послідовності, які діляться націло на 3 : вивести такі числа;
 - (б) кількість чисел з послідовності, які діляться на 3 з остачею 1 : вивести такі числа;
 - (с) кількість чисел з послідовності, які діляться на 3 з остачею 2 : вивести такі числа;
 - (д) шлях до каталогу, у якому міститься файл, час створення і розмір файлу.
2. У вхідному файлі `data2.txt` міститься графік руху потягів між двома станціями. Дозаписати у цей файл рядки з впорядкованим за часом відправлення графіком руху потягів та доданою інформацією для кожного потягу про його час у дорозі.
Приклад файлу `data2.txt`
1 11:26 17:22
2 19:04 01:36
3 16:12 23:08
3. Задано словник з N англomовних слів без повторів слів. Виконати такі завдання:
 - (а) серіалізувати цей словник у бінарний файл `binary.data`;
 - (б) десеріалізувати дані з файлу і вивести на екран кількість слів у словнику, що починаються введеною буквою, і самі слова.

IV. БОНУСНЕ ЗАВДАННЯ

У кожної кредитної картки є надрукований на ній номер, а також номер, вбудований у магнітну стрічку зі зворотного боку картки. Останній зберігається також у базі даних для того, щоб коли картою розраховуються за покупку, банк знав, з якого рахунку списувати гроші. У світі багато людей з кредитними картками, тому ці номери доволі великі: American Express використовує номери з 15 розрядів, MasterCard — з 16, Visa — з 13 або 16. Ці розряди — десяткові (від 0 до 9). Номери кредитних карток мають певну структуру. Номери карток American Express починаються з 34 чи 37; номери карток MasterCard починаються з 51, 52, 53, 54 чи 55; номери карток Visa починаються з 4. Також номери карток мають вбудовану у них контрольну суму, що обчислюється за допомогою математичної формули, використовуючи алгоритм Ганса Луна з IBM. Відповідно до цього алгоритму, можна перевірити, чи номер кредитної картки є синтаксично коректним, наступним чином: рядки довжиною 1 або менше не є дійсними; пропуски дозволені для введення, але їх необхідно видалити перед перевіркою; усі інші нецифрові символи заборонені.

Приклад 1: дійсний номер кредитної картки 4539148803436467

Перший крок алгоритму Луна — це подвоєння кожної другої цифри, починаючи справа: 4_3_1_8_0_4_6_6_

Якщо подвоєння числа є числом, що перевищує 9, необхідно відняти 9 від добутку.

Результати подвоєння розставлені на свої місця: 8569 2478 0383 3437

Далі необхідно підсумувати всі цифри: $8+5+6+9+2+4+7+8+0+3+8+3+3+4+3+7 = 80$

Якщо сума націло ділиться на 10, то номер картки дійсний. Цей номер дійсний!

Приклад 2: недійсний номер кредитної картки 8273123273520569

Подвоєння кожної другої цифри, починаючи справа: 7253226253120539

Сума цифр: $7+2+5+3+2+2+6+2+5+3+1+2+0+5+3+9 = 57$

57 націло не ділиться на 10, тому цей номер картки недійсний!

Кожен рядок вхідного файлу numbers.txt містить номер кредитної картки. Напишіть функцію(ї) для валідації номерів кредитних карток за алгоритмом Луна та виведення приналежності картки до трьох компаній (American Express, MasterCard, Visa), або повідомлення INVALID — в іншому разі.

ТЕСТ ВХІД:

4003600000000014

5105105105105100

378282246310005

4111111111111113

ВИХІД:

Visa

MasterCard

American Express

INVALID

V. ЗВІТ ПО ЛАБОРАТОРНІЙ РОБОТІ

Зміст звіту:

1. Титульний аркуш зі стандартними атрибутами
2. Коди програм з коментарями
3. Тестування програм
4. Висновки

СПИСОК ЛІТЕРАТУРИ

- [1] Васильєв О.М. Програмування мовою Python. – Тернопіль: Навчальна книга — Богдан, 2019.
- [2] Висоцька В. А., Оборська О. В. Python: алгоритмізація та програмування: навчальний посібник — Львів: Новий Світ — 2000, 2021.
- [3] Крєневич А. П. Python у прикладах і задачах. Частина 1. Структурне програмування: Навчальний посібник. – К.: ВПЦ "Київський університет", 2017.
- [4] Мізюк О. Путівник мовою програмування Python. – Ел. вид., 2020. – Режим доступу: <https://pythonguide.rozh2sch.org.ua>
- [5] Селіверстов Р., Мельничин А. Основи програмування мовою Python: навч. посібник. — Львів: ЛНУ імені Івана Франка, 2020.
- [6] Severance Ch. R. Python for Everybody: Exploring Data Using Python 3, 2023. – Access mode: https://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf
- [7] Swaroop С. Н. A byte of Python, 2005. – Access mode: <https://python.swaroopch.com>
- [8] PEP 8 — Style Guide for Python Code. – Access mode: <https://peps.python.org/pep-0008/#source-file-encoding>
- [9] The Python Tutorial. – Access mode: <https://docs.python.org/3/tutorial/index.html>