

Лабораторна робота № 3

ЗАГАЛЬНА ПОСТАНОВКА ЗАДАЧІ.....	1
Уточнення загальної постановки задачі	1
ВИМОГИ.....	1
АЛГОРИТМ РОБОТИ ПРОГРАМИ.....	1
СТРУКТУРА ПРОГРАМИ.....	2
ІНШЕ	2
ОФОРМЛЕННЯ КОДУ	3
ПРИНЦИПИ АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ (ДУЖЕ ВАЖЛИВО!).....	3
ЗВІТНІСТЬ	3
ОРГАНІЗАЦІЙНІ МОМЕНТИ.....	3
ПОПЕРЕДНЯ ПРОГРАМНА ПЕРЕВІРКА (INF)	4
ПЕРЕВІРКА ЛАБОРАТОРНИХ РОБІТ	4

Загальна постановка задачі

Написати програму для розв'язання такої задачі (по варіантах).

Приклад варіанту задачі. З послідовності цілих вилучили всі числа, що діляться на 3. Знайти першу найдовшу підпослідовність залишку, що йде поспіль (у залишку) та строго зростає.

За послідовності 11 6 17 3 22 4 8 10 має бути знайдено 11 17 22 .

Уточнення загальної постановки задачі

Напишіть програму, яка запитує в користувача послідовність цілих та у відповідь виводить введену послідовність та знайдену підпослідовність.

На початку роботи програми має виводитися інформація щодо виконавця, стисла умова задачі та номер варіанту (друкувати формули НЕ треба).

Варіанти завдань та розподіл варіантів (такий самий, як для ЛР1) – див. файли

lab3.pdf та **Variants.xlsx** відповідно.

Жодних друкованих/рукописних папірців здавати **НЕ ТРЕБА**.

Вимоги

Наведені далі вимоги описують уточнення постановки задачі та деталі оформлення лабораторної роботи. Рекомендовано не починати кодування до тих пір, поки цей документ не переглянуто до кінця хоча б по діагоналі.

Позначка (**inf**) означає, що даний пункт є інформативним та на оцінювання не впливає. Позначка (**F**) позначає, що виконання цієї вимоги впливатиме на оцінювання якості коду. Якщо позначок нема, то вимога є **критичною** і її варто виконати.

Якщо не зазначено інше, під наявністю функції/методу/класу/... мається на увазі наявність належно реалізованої функції/..., що відповідає умові лабораторної та вимогам до неї.

Алгоритм роботи програми

A1. Виконання програми за коректних вхідних даних відповідає такому алгоритму (зі збереженням хронології):

- 1) повідомити виконавця;
- 2) повідомити призначення програми (номер варіанту та стисло умову задачі, формулу не виводити);
- 3) ввести оброблювану послідовність;
- 4) знайти вказану в умові підпослідовність;
- 5) з нового рядка вивести рядок
******* Input**
- 6) з нового рядка вивести вхідну послідовність;

7) з нового рядка вивести рядок

***** **Output**

8) з нового рядка вивести знайдену підпоследовність;

9) з нового рядка вивести рядок

Виділені частини вихідного тексту мають виводитися з 100%-ю точністю.

A2. **Введення послідовності** завершується за появою винятку **KeyboardInterrupt**.

Числа можуть вводиться по кілька на рядок, розділені білими символами. Може бути введено кілька рядків. За помилки перетворення, вхідна лексема, на якій виникла помилка, ігнорується та процес введення продовжується з наступної за неї лексеми. Наприклад, за рядка

12 123у 45

до послідовності мають бути додані цілі 12 та 45.

Запитувати в користувача довжину послідовності заборонено.

A3. **Виведення послідовності** (та її підпоследовності) здійснюється через кому, у кінці послідовності значень, якщо вона непорожня, ставиться крапка й виконується перехід на новий рядок. За порожньої послідовності виводиться повідомлення **Empty**, і також виконується перехід на новий рядок.

A4. (F) Наявні зрозумілі підказки на введення даних.

A5. Програма правильно знаходить підпоследовність відповідно до варіанту. Наприкінці у зрозумілому вигляді виводить введену послідовність та знайдену підпоследовність.

A6. Програма правильно визначає неможливість обчислення, якщо задані вхідні дані не належать області визначення виразу, і друкує повідомлення про помилку.

A7. Аварійне завершення програми є неприпустимим.

A 8. Програма має передбачати обробку послідовностей довільної довжини (у межах, що підтримуються інтерпретатором мови Python).

Структура програми

S1. Програма записана рівно в одному текстовому файлі з розширенням **py**, в імені файлу використовуються виключно символи US-ASCII. У ній відсутні синтаксичні помилки та вона може бути виконана інтерпретатором Python версії 3.8.

S2. **Структура даних для збереження вхідної послідовності**. Вхідна послідовність має зберігатися в списку. Цей список є **єдиною** використовуваною структурою даних для збереження вхідної послідовності чи її частин.

S3. **Обмеження на використання пам'яті**. Заборонено використовувати додаткові допоміжні структури даних або зовнішні носії для збереження вхідної послідовності або її підпоследовностей (як елементів, так й їхніх номерів, а також перетворених на рядковий тип частин послідовності).

S4. **Обмеження на час роботи**. Програма може виконувати не більше чотирьох проходів по списку: перший – під час заповнення, другий – під час пошуку підпоследовності, третій – під час виведення вхідної послідовності, четвертий – під час виведення знайденої підпоследовності.

S5. Інструкції **global**, **nonlocal**, **break** заборонені до використання.

S6. Завершення функції за інструкцією **return** з середини циклу заборонено.

S7. Власні глобальні змінні використовувати заборонено.

S8. Заборонено використовувати вкладені та/або рекурсивні функції.

S9. Усі функції означені на рівні модуля. Інструкції означення функцій не повинні входити до складу інших інструкцій.

S10. Інструкції імпортування відсутні.

Інше

M1. Усі надіслані версії лабораторної не порушують принципи академічної доброчесності.

M2. Усі ідентифікатори записані символами US-ASCII.

M3. (F) Код не повинен містити дублювання та невикористовувані фрагменти (навіть закоментовані).

M4. (F) Коментарі не повинні бути надлишковими.

M5. Використання у власному коді **exit** або інших засобів дострокового завершення програми заборонено.

M6. Використання **async** не дозволяється. (Хоча це неймовірно, щоб в цій лабці воно комусь дійсно було потрібно.)

M7. Код має відповідати рекомендаціям підрозділів 7.1.6 та 7.4 (див. файл **7 Керування порядком обчислень.pdf**).

M8. (F) Використовувані в програмі імена є змістовними, їх призначення зрозуміле з їх назви. Довгих імен слід уникати.

M9. (F) Назви суто допоміжних функцій мають починатися з підкреслення. Усі функції належно задокументовані.

M10. (F) Під час проектування слід дотримуватись принципів функціонального проектування, уникати довгих функцій.

M11. (F) Код не має бути заплутаним. Код не має бути захарашений перевірками. Використання механізму винятків має бути доречним та адекватним.

M12. (F) Власних функцій, що повертають рядки, бути не повинно.

M13. Змінні, що використовуються в тілі функції виключно як локальні змінні функції для збереження проміжних результатів, не повинні бути її параметрами.

M14. Кожен фізичний рядок файлу з текстом програми цілком вміщується в одну ширину екрана ноутбука. (Текст програми можна прочитати в текстовому редакторі без горизонтальної прокрутки.)

Оформлення коду

Документацію та коментарі краще писати англійською мовою (PEP8). Якщо це ну дуже важко, то за використання інших мов бали знижуватися не будуть.

Для файлів з текстом програми використовується виключно кодування **UTF-8** (це основне використовуване Python кодування).

Принципи академічної доброчесності (дуже важливо!)

Студент вільно орієнтується в коді лабораторної роботи, яку він здає, розуміє усі використані синтаксичні елементи мови та бібліотечні засоби, зміст та призначення частин коду, вміє самостійно запустити програму на виконання, здатен самостійно внести неглобальні виправлення в код. У коді відсутні коментарі, що перекладають зміст інструкцій на слов'янські мови.

Принципи академічної доброчесності передбачають, що ані брати чужий код, ані давати комусь свій не можна. Сумісна розробка лабораторних також заборонена.

Якщо листи з кодом та скріншотами лабораторної відправлено не з власної поштової адреси (а з адреси товариша!), або якщо з однієї адреси відправлено лабораторні різних студентів, або якщо студент помилився в номері варіанту, або виконав не свій варіант, або неправильно зазначив виконавця, або виконавцем зазначений хтось інший, то все це також вважається порушенням академічної доброчесності.

Звітність

Звітністю з цієї лабораторної є: розташована в коді документація та надісланий код.

Жодних друкованих/рукописних папірців окремо оформлювати та здавати **НЕ ТРЕБА**.

Організаційні моменти

Не пізніше 1 грудня 2020 року на адресу LabAssignment2@i.ua (далі «адреса для лабораторних робіт») надійшов лист з повним кодом лабораторної роботи (усі суттєві для проекту ру-файли і нічого іншого, як вкладення). У темі листа зазначено, що це лабораторна робота 2 та номер варіанту, у форматі **Lab3, <номер варіанту>**. Наприклад, **Lab3, 66**

У самому листі зазначено виконавця та середовище, що використовувалося для виконання лабораторної роботи. Листи з архівами та посиланнями на інтернет-ресурси не припустимі. Один лист – одна лабораторна робота, повністю.

Дозволяється відправляти код лабораторної кілька разів (наприклад, якщо було усунуто якісь недоліки). У випадку, коли код лабораторної надходив кілька разів, розглядатиметься та

оцінюється тільки остання версія (навіть, якщо передостання працювала, а остання "зламалася"). Дата/час версії визначається за датою надходження на адресу для лабораторних робіт. Кількість спроб на оцінювання не впливає.

Попередня програмна перевірка (inf)

Не планується.

Перевірка лабораторних робіт

Перевірка лабораторних робіт розпочнеться ПІСЛЯ завершення терміну приймання.

Максимально можлива кількість балів за лабораторну становить 20 балів (з 100 за семестр).

Розв'язувана задача має відповідати умові та варіанту.

Отримана сукупність лабораторних робіт ПРОГРАМНО перевіряється на дотримання принципів академічної доброчесності (запозичення коду). Чим більше буде збіг, тим більш ретельною буде співбесіда. Якщо збіг буде зашкалювати, то всі такі лабораторні будуть оцінені в 0 балів, незалежно від того, хто був клієнтом, а хто сервером.

Не таємниця, що умови майже однакові і на лабораторних заняттях схожі задачі розв'язуються. Але власноруч набраний код, навіть за мотивами розібраної схожої задачі, буде набагато більш різноманітним, ніж код, отриманий заміною деяких символів у набраному кимось файлі. І програма це чудово вміє враховувати, щоб визначити код, який був переважно просто скопійований.

Лабораторна робота передбачає захист у формі співбесіди.

Співбесіда за лабораторною (у zoom) може відбуватися не тотально, а тільки в тих випадках, коли наявна інформація (інші результати оцінювання, надісланий код та скріншоти, відгуки викладачів, що ведуть лабораторні заняття; результати програмної перевірки на запозичення) або її відсутність ставить під сумнів дотримання принципів академічної доброчесності під час виконання лабораторної роботи.

Захист лабораторної роботи вважається неуспішним, якщо під час захисту виявляється, що студент не до кінця розуміє код або погано в ньому орієнтується чи не розуміє використані синтаксичні елементи мови, зміст та призначення частин коду, а також якщо захист не відбувся з ініціативи студента.

IF є порушення критичних вимог:

лабораторна отримує 0 балів

ELSEIF наявні недоліки, пов'язані з якістю коду та порушеннями вимог категорії **F**:

лабораторна отримує 15 балів

ELSE:

лабораторна отримує 20 балів

Бажаю всім потрапити в останню гілку обчислень (і без співбесіди)!

Порада. Перед тим, як надсилати код лабораторної, варто її ретельно самостійно **протестувати**, а також з олівцем перевірити виконання вимог (методом: перевірили, якщо виконується, поставили галочку). За необхідністю внести зміни і ще раз **протестувати**. І якщо все гаразд, то лабораторну можна здавати ☺.