

Контрольна лабораторна робота № 1

ЗАГАЛЬНА ПОСТАНОВКА ЗАДАЧІ..... 1

ВИМОГИ..... 1

АЛГОРИТМ РОБОТИ ПРОГРАМИ	1
КОНТРОЛЬ ПОМИЛОК ВВЕДЕННЯ.....	2
СТРУКТУРА ПРОГРАМИ.....	2
ІНШІ ОБМЕЖЕННЯ.....	3
ОФОРМЛЕННЯ КОДУ	3
ОРГАНІЗАЦІЙНІ МОМЕНТИ.....	3
ЗВІТНІСТЬ	4
АКАДЕМІЧНА ПОРЯДНІСТЬ (ДУЖЕ ВАЖЛИВО!)	4

ПЕРЕВІРКА ЛАБОРАТОРНИХ РОБІТ 4

Загальна постановка задачі

Напишіть програму, яка запитує в користувача значення дійсної змінної та у відповідь друкує введене значення змінної та значення виразу для цього значення. Для вхідних даних, що не належать області визначення виразу, має друкуватися відповідне повідомлення про неможливість обчислень.

На початку роботи програми має виводитися умова задачі, номер варіанту та інформація щодо виконавця.

Варіанти завдань та розподіл варіантів – див. файли **lab1.pdf** та **Варіанти лабораторних.xls** відповідно.

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.

Увага! Планується автоматизоване тестування (і як «чорної» скриньки, і як «білої» скриньки). Тестер доволі чутливий до порушення вимог. Прошу відповідально віднестися до дотримання усіх вимог до лабораторної.

Позначка **(inf)** означає, що даний пункт є інформативним та на оцінювання не впливає. Позначка **(F)** позначає, що виконання цієї вимоги впливатиме тільки на оцінювання якості коду. Якщо позначок нема, то вимога є **критичною**: у випадку порушення лабораторна робота оцінюється в **0 (нуль)** балів.

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 відповідає такому алгоритму (зі збереженням хронології):

1) повідомити призначення програми (номер варіанту та стисло умову задачі, формулу не виводити);

2) повідомити виконавця;

3) ввести вхідні дані;

4) з нового рядка вивести

******* do calculations ...**

5) провести обчислення;

6) вивести

done

У результаті на екрані маємо бачити рядок

******* do calculations ... done**

7) з нового рядка вивести введені дані (у форматі з фіксованою крапкою, 6 знаків після коми, число взято як приклад)

```
for x = 11.222090
```

8) на наступному рядку вивести результати обчислення (у форматі з фіксованою крапкою, 6 знаків після крапки, число взято як приклад)

```
result = 13.131313
```

або

```
result = undefined
```

у залежності від того, чи можна обчислити вказану у варіанті функцію для введеного значення.

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

A2. Значення змінної вводиться користувачем. На введення змінної користувачу дається рівно одна спроба! (Повторні запити на введення категорично забороняються.)

A3. (F) Наявні зрозумілі підказки на введення даних.

A4. Програма правильно виконує математичні обчислення відповідно до варіанту, якщо введене значення належить області визначення виразу.

A5. Програма правильно визначає неможливість обчислення, якщо задані вхідні дані не належать області визначення виразу.

A6. Наприкінці у зрозумілому вигляді виводяться введені дані та результат обчислення для них (неможливість обчислення – теж результат ☺).

A7. (inf) Більш детальна діагностика причини неможливості обчислення може бути виведена за бажанням виконавця, але тільки в самому кінці програми (після п.8 алгоритму).

A8. Інструкції, що відповідають за виведення та обчислення, розташовуються послідовно (спочатку обчислення, потім виведення) й **не** перетинаються в тексті.

A9. Аварійне завершення програми є неприпустимим.

Контроль помилок введення

E1. Помилки введення (введення нечислових значень) мають контролюватися. У випадку, коли користувач ввів послідовність символів, що не може бути перетворена на тип **float**, після п.3 алгоритму на окремому рядку має бути виведено

```
wrong input
```

Далі за бажанням виконавця з нового рядка може бути ще щось виведено (але не про введене значення та не про результат обчислень на ньому).

На цьому програма завершує роботу (обчислення не виконуються, повторний запит на введення не дається).

Структура програми

S1. Програма записана рівно в одному текстовому файлі з розширенням **py**. У ній відсутні синтаксичні помилки та вона може бути виконана інтерпретатором Python версії 3.7.4.

S2. Наявні функції з іменами **domain**, **f**, **f_total**.

Функція **domain** за дійсним аргументом має перевіряти його належність області визначення виразу; повертає значення, яке має оцінюватися як істина, якщо аргумент належить області визначення. (Зверніть увагу: вимога, щоб ця функція повертала значення типу **bool** не ставиться.)

Функція **f** приймає один дійсний аргумент, за яким виконує обчислення виразу і повертає обчислене значення. Жодних перевірок не виконує.

Функція **f_total** приймає один дійсний аргумент, за яким виконує обчислення виразу, тільки якщо це можливо. Для перевірки застосовності використовує функцію **domain**. В якості результату повертає пару значень: булівську ознаку успішності обчислення (True, якщо обчислення виконане, False – інакше) та дійсне – результат обчислення. За неуспішного обчислення, друга компонента результату значення не має.

S3. Жодна з функцій **domain**, **f**, **f_total** **НЕ** спілкується з користувачем, нічого не знає про консоль, не використовує глобальні змінні.

S4. (F) Усі функції належно задокументовані.

S5. Функції **domain** та **f** можуть використовуватися тільки у **f_total**.

Інші обмеження

M1. Під час обчислень використовуються значення математичних констант π та e (якщо вони наявні в заданій формулі) з бібліотеки **math**. Крім **math** та його засобів нічого іншого не імпортується.

M2. (F) Можна написати одну функцію без аргументів, яка буде виконувати завдання лабораторної «під ключ», тобто виконувати запит значення, обчислення та повідомлення результату. У такому випадку глобальних змінних бути не повинно. За відсутності такої функції програма може використовувати не більше трьох глобальних змінних: вхідне значення, ознака успішності обчислення, результат обчислення.

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

M4. (F) Коментарі не повинні бути надлишковими.

M5. Використання у власному коді **exit** заборонено.

M6. Код має відповідати рекомендаціям підрозділів 7.1.6 та 7.4 (див. файл **07 Керування порядком обчислень.pdf**).

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).

Організаційні моменти

Не пізніше 22 листопада 2019 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 1 та номер варіанту, у форматі **Lab1, <номер варіанту>**. Наприклад, **Lab1, 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 до останньої забули додати частину файлів). Дата/час версії визначається за датою надходження на адресу для лабораторних робіт. Кількість спроб на оцінювання не впливає.

Не пізніше 24 листопада 2019 року на адресу LabAssignment2@i.ua надійшов лист із скріншотами (jpeg, розмір зменшити, додати до листа як вкладення), що відображають:

- файли програми з вхідним кодом (у менеджері проектів або текстовому редакторі відповідного середовища),

- виконання програми (достатньо показати консоль на момент, коли вже програма допрацювала до кінця; для випадку коректних та некоректних (не з ОДЗ або нечислових) вхідних даних).

У першому випадку в текстовому редакторі використовуваного середовища має бути відкрито файл з текстом програми та має бути помітно рядки цього файлу.

У другому випадку з скріншота повинно бути видно у який спосіб виконувалася програма: з-під середовища чи безпосередньо з командного рядка.

Оформлення листа аналогічне попередньому: тема листа аналогічна темі листа з кодом лабораторної, у самому листі зазначено виконавця та середовище, що використовувалося при виконанні лабораторної роботи. Листи з архівами та посиланнями на інтернет-ресурси не припустимі.

Скріншоти мають відображати останню надіслану версію лабораторної роботи!

Оскільки виправлення коду допускаються, рекомендовано надсилати скріншоти тільки тоді, коли є повна впевненість, що код більше змінюватися не буде.

Код та скріншоти надсилати різними листами!

Звітність

Звітністю з цієї лабораторної є: розташована в кодї документація, надісланий код та надіслані скріншоти з виконанням тестів.

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

Академічна порядність (дуже важливо!)

Студент вільно орієнтується в кодї лабораторної роботи, яку він здає, розуміє використані синтаксичні елементи мови, зміст та призначення частин коду, вміє самостійно запустити програму на виконання.

Використання чужих файлів з кодом категорично заборонено.

Усі листи з кодом та скріншотами лабораторної відправлено із власної поштової адреси (а не з адреси товариша!). У випадку, коли з однієї адреси відправлено лабораторні різних студентів, всі такі лабораторні будуть оцінені на **0 (нуль)** балів.

Перевірка лабораторних робіт

Максимально можлива кількість балів за лабораторну становить 16 балів (з 100 за семестр).

Скріншоти є невід'ємною частиною лабораторної роботи. За їх відсутності лабораторна робота не вважається виконаною. Розв'язувана задача має відповідати умові та варіанту.

Оцінювання коду лабораторної роботи складається з трьох етапів:

1) на першому етапі код лабораторної роботи, що був наданий на перевірку в електронному вигляді за визначеним в умові лабораторної роботи протоколом, проходить тестування; якщо хоча б один тест не пройдено, лабораторна отримує 0 балів.

2) на другому етапі відбувається перевірка якості коду (за наявності недоліків оцінка знижується на 25%);

3) на третьому етапі відбувається захист лабораторної роботи (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, то лабораторна робота отримує нуль балів).

Перший етап перевірки виконується програмою, яка перевіряє лабораторну і як «чорну» скриньку, і як «білу» скриньку. Програма орієнтується на вимоги, що зазначені вище в цьому файлі. Вона перевіряє не тільки математичну сторону, але й організацію коду програми. Тестер є критично чутливим до назв функцій, розташування вихідного тексту по рядках, виділених жовтим текстових повідомлень, структури алгоритму, послідовності обчислень/виведень. Спроба написати програму якимось інакше може призвести до того, що код банально не пройде тестування.

У більшості випадків тестер працює до першого порушення. Тому розраховувати на те, що він видасть повну діагностику, не варто. Хоча кількість спроб не обмежується, але може так статися, що тестер буде запускатися 2-3-4 рази на тиждень. Якщо кожного разу робити тільки по одному виправленню, то можна буде не встигнути до 22.11.19 виправити все.

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

Якщо тестер надіслав негативну відповідь, то варто її прочитати та також ретельно перечитати та перевірити виконання вимог.

Далі перевірятиметься те, що недоперевірів тестер, та якість коду належно працюючої програми. Переважно перевірятимуться вимоги, позначені як **(F)**, а також стиль використання винятків.

Співбесіда за лабораторною буде відбуватися тільки в тих випадках, коли наявна інформація (інші результати оцінювання, надісланий код та скріншоти) ставить під сумнів дотримання принципів академічної доброчесності під час виконання лабораторної роботи.