

## Контрольна лабораторна робота № 1

### ЗАГАЛЬНА ПОСТАНОВКА ЗАДАЧІ..... 1

### ВИМОГИ..... 1

АЛГОРИТМ РОБОТИ ПРОГРАМИ.....	1
КОНТРОЛЬ ПОМИЛОК ВВЕДЕННЯ.....	2
СТРУКТУРА ПРОГРАМИ.....	2
ІНШЕ.....	3
ОФОРМЛЕННЯ КОДУ.....	3
ПРИНЦИПИ АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ (ДУЖЕ ВАЖЛИВО!).....	3
ОРГАНІЗАЦІЙНІ МОМЕНТИ.....	4
ЗВІТНІСТЬ.....	4
ПОПЕРЕДНЯ ПРОГРАМНА ПЕРЕВІРКА (INF).....	5

### ПЕРЕВІРКА ЛАБОРАТОРНИХ РОБІТ ..... 6

#### Загальна постановка задачі

Напишіть програму, яка запитує в користувача значення дійсної змінної та у відповідь друкує введені 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 інформація щодо виконавця та умова задачі, номер варіанту.

Варіанти завдань та розподіл варіантів – див. файли **lab1.pdf** та **Variants.xlsx** відповідно. Якщо хтось не знайшов себе у файлі з варіантами, терміново звертайтеся. Списки студентів були отримані з деканату на самому початку семестру, можливі неточності.

#### 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 відповідає такому алгоритму (зі збереженням хронології):

- 1) повідомити виконавця;
- 2) повідомити призначення програми (номер варіанту та стисло умову задачі, формулу не виводити);
- 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, 8 знаків після коми, число взято як приклад)  
**for x = 11.22209011**

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

```
result = 13.13131311
```

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%-ю точністю.

Вихід програми має приблизно такий вигляд.

```
This program is coded by ...
```

```
This program calculates .....
```

```
Enter something...
```

```
...
```

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

```
for x = 11.22209011
```

```
result = 13.13131311
```

```
...
```

A2. Значення змінної вводиться користувачем. На введення змінної користувачу дається рівно одна спроба! (Повторні запити на введення категорично забороняються. Крім введення значення змінної жодних інших запитів на введення або натискання клавіш у програмі нема!)

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 в 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 усіх "магічних" фраз) і з нового рядка.

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

A9. Аварійне завершення програми є неприпустимим.

### Контроль помилок введення

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

```
wrong input
```

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**, в імені файлу використовуються виключно символи US-ASCII. У ній відсутні синтаксичні помилки та вона може бути виконана інтерпретатором Python версії 3.8.

S2. Наявна функція з іменем **f**, що приймає один дійсний аргумент, за яким виконує обчислення виразу і повертає обчислене дійсне значення. Жодних перевірок не виконує. Обчислення є її єдиною відповідальністю. Викликається рівно один раз. Крім як у функції **f**, ніде більше вирази, що явно задають обчислення за заданою в умові формулою або її частинами, не записані.

S3. Крім функції **f** мають бути наявні інші функції.

S4. Власні глобальні змінні використовувати заборонено.

S5. Усі функції означені на рівні модуля.

S6. Інструкції **try** не повинні бути вкладеними (за текстом) в інші інструкції **try**.

S7. Під час обчислень використовуються значення математичних констант  $\pi$  та  $e$  (якщо вони наявні в заданій формулі) з бібліотеки **math**. Інструкція імпортування має бути записана так і тільки так:

```
import math
```

Крім **math** нічого іншого не імпортується.

### **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 захаращений перевітками. Використання механізму винятків має бути доречним та адекватним.

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

M13. Рядки в операторах порівняння не порівнюються. (Це насправді більше підказка. Якщо в цій задачі десь виникає порівняння рядків, то явно щось не так спроектовано.)

M14. Змінні, що використовуються в її тілі виключно як локальні змінні функції для збереження проміжних результатів, не повинні бути її параметрами.

M15. 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.

## Організаційні моменти

Не пізніше 15 листопада 2020 року на адресу [LabAssignment2@i.ua](mailto: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 остання "зламалася"). Дата/час версії визначається за датою надходження на адресу для лабораторних робіт. Кількість спроб на оцінювання не впливає.

Не пізніше 17 листопада 2020 року на адресу [LabAssignment2@i.ua](mailto: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 коді документація, надісланий код та надіслані скріншоти з виконанням тестів.

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

### Попередня програмна перевірка (inf)

Ця задача насправді нескладна і може бути запрограмована кількома шаблонами. У залежності від обраного шаблону, у всій програмі достатньо не більше 2 інструкцій **if** та не більше 2 інструкцій **try** (якихось з них, можливо, навіть менше). Менше двох функцій точно бути не повинно. Але за рівно двох функцій якась з них з може виявитись довгою.

Статистика каже, що більшість студентів успішно виконують цю лабораторну 😊.

Щоб мінімізувати кількість зовсім поганих балів, програмно перевіряється виконання дуже невеликої частини критичних вимог (переважно арифметика), протокол перевірки повідомляється. Ця попередня перевірка буде розпочата майже відразу, як почнуть надходити лабораторні роботи.

**Більшість критичних вимог перевірятись програмно не буде** (нема часу програмувати ці перевірки)! Іноді тестер може давати підказки щодо потенційних порушень (можливо, що порушення є, можливо, що нема). Їх варто прочитати.

Звертаю увагу, що є частина вимог, пов'язана з оформленням і через порушення яких тестер не зможе обробити суть та буде писати на перший погляд дивні речі. Тут 1) **тестер правий, що відхиляє лабораторну, бо він працює згідно загальних вимог, які зазначені в цьому файлі**; 2) варто перевіряти, чи в тому вигляді, послідовності, форматі виводиться те, що має виводитись.

Щодо критичних вимог, які тестер перевіряти не буде, та якості коду, то є викладачі, що ведуть лабораторні заняття. Не найгірша думка інколи з ними консультуватись.

Якщо раптом виявляється, що код стає заплутаним, у ньому з'являються різноманітні "затички", збільшується кількість перевірок та розгалужень, то це явна ознака того, що щось явно не так.

## Перевірка лабораторних робіт

Перевірка лабораторних робіт розпочнеться ПІСЛЯ завершення терміну приймання.

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

Скріншоти є невід'ємною частиною лабораторної роботи. За їх відсутності/невідповідності лабораторна робота не вважається виконаною. 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** (тестер відхилив лабораторну,

або хоча б одна версія порушує принципи академічної доброчесності,

або лабораторна має критичну кількість запозичень,

або порушено S2,

або є три чи більше порушень критичних вимог,

або захист був неуспішним ) :

лабораторна отримує 0 балів

**ELSEIF** є 1-2 порушення критичних вимог:

лабораторна отримує 12 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 ☺.