

ПРАКТИЧНА РОБОТА 11

Функціональне програмування в Python

Система оцінювання

№	Тема	К-ть балів
1.	<i>Захист принаймні одного завдання з роботи</i>	1
2.	Практичні завдання	3,6*
3.	<i>Здача звіту</i>	0,4
	Всього	5

* – діє бонусна система

- 0,3 бала (**Спискове включення**) Функція foo(C) потрібна, щоб повернути «зворотну» версію символьного рядка:

```
>>> foo("hawk")
'kwah'
>>> foo("amazing")
'gnizama'
```

Напишіть таку функцію відповідно до наступної стратегії. Спочатку перевірте, чи можете написати range-вираз, який проходить від індексу останньої букви до першого індексу. Наприклад, для рядка "thousand" range-вираз має повернути [7,6,5,4,3,2,1,0]. Використайте цей вираз, щоб зробити власний список, який виконуватиме схожий код: [C[i] for i in range(. . .)]
 Це допоможе спростити foo(C), яка повертатиме цей кастомний список. Не використовуйте reverse(). Залишіться перетворити список символів у рядок. Використовуйте для цього метод join().
- 0,3 бала (**Лямбди та включення**) Напишіть анонімну функцію, яка видалятиме голосні букви із введеного Вами тексту.
- 0,4 бала (**Ітератори**) Напишіть власний нескінченний ітератор. Внутрішні обчислення для формування нових елементів послідовності обирайте на Ваш розсуд.
- 0,4 бала (**Генератори**) Напишіть функції-генератори, які дозволятимуть обчислювати n -тий член ланцюга [Каннінгема](#) першого та другого роду.
- 0,4 бала (**Декоратори**) Напишіть власний декоратор, який обрамлятиме введений текст символами, переданими в декоратор. Наприклад, для знаку = та введеного тексту
 Привіт, світ
 виведе
 =====
 = Привіт, світ =
 =====

6. *0,6 бала* **(Функції вищого порядку)** Напишіть власні реалізації функцій вищого порядку reduce, map, filter. Створіть декоратор, який повертатиме час роботи функцій та порівняйте часи роботи ваших реалізацій та вбудованих версій reduce, map, filter. Функцію, до якої застосовуватиметься код, пиште на власний розсуд, проте зверніть увагу, щоб вона виконувалась відносно довго.
7. *0,6 бала* Маючи список з 10 GPS-координат, визначте пару найближчих положень. Відстань між двома розташуваннями слід визначати за [формулою гаверсинуса](#).
8. *0,6 бала* Створіть словник з абонентами та їх номерами мобільних телефонів. Використовуючи функціональний стиль програмування, виведіть списки абонентів відповідно до [коду](#) їх мобільного оператора.