

## Завдання №1

Ваша програма має бути записана в окремому файлі, який може виконуватися консольною командою `python <ім'я файлу>.py`.

Результат обчислень має виводитися на екран командою `print`.

Кожна програма має сприймати довільні значення, які задовольняють описаному в завданні формату, і передаються у вигляді аргументів командного рядка при виклику програми: `python <ім'я файлу>.py arg1 arg2 arg3`

Формат введення даних та виведення результатів має ТОЧНО збігатися із вказаним.

Складіть код, перевірте його на різних прикладах. Зверніть увагу, що тестові приклади до завдань лише демонструють логіку роботи та показують формат даних.

**1)Вхідні дані:** 2 невід'ємних дійсних числа  $a$  та  $b$  -- аргументи командного рядка.  $b$  не дорівнює 0.

**Вихідні дані:** дійсне число -- результат обчислення формули

$$x = \frac{\sqrt{ab}}{e^a * b} + ae^{\frac{2a}{b}}$$

### Приклад

Вхідні дані: 0 1

Приклад виклику: `python test.py 0 1`

Результат: 0.0

Вхідні дані: 0.5 10

Приклад виклику: `python test.py 0.5 10`

Результат: 0.688209837593

**2)Вхідні дані:** 3 дійсних числа -- аргументи командного рядка.

**Вихідні дані:** результат обчислення формули

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

Аргументи передаються в порядку, зазначеному у формулі, назви змінних можуть використовуватися будь-які.

### Приклад

Вхідні дані: 1 1 0.25

Приклад виклику: `python lab2_1.py 1 1 0.25`

Результат: 1.59576912161

**3)Вхідні дані:** 3 числа  $x$ ,  $y$  та  $z$ .  $x$ ,  $y$  -- невід'ємні цілі числа,  $z$  дорівнює 0 або 1.  $x$  не дорівнює 0. Передаються як аргументи командного рядка.

**Вихідні дані:** рядок "Everybody sing a song: <текст пісеньки>.", де <текст пісеньки> формується з  $y$  куплетів, розділених пробілами. Всі куплети однакові і складаються з  $x$  'la' через дефіс. Якщо  $z$  дорівнює одиниці, в кінці ставиться окличний знак, інакше крапка. За відсутності куплетів пробіл перед крапкою/окличним знаком не ставиться.

Підказка: для цього можна застосувати множення рядків.

### Приклад

Вхідні дані: 2 3 1

Приклад виклику: `python lab2_2.py 2 3 1`

Результат: Everybody sing a song: la-la la-la la-la!

Вхідні дані: 1 0 0

Приклад виклику: `python lab2_2.py 1 0 0`

Результат: Everybody sing a song:.

### Завдання №2

**1)Вхідні дані:** 3 дійсних числа  $a$ ,  $b$ ,  $c$ . Передаються в програму як аргументи командного рядка.

**Результат роботи:** рядок "triangle", якщо можуть існувати відрізки з такою довжиною та з них можна скласти трикутник, або "not triangle" -- якщо ні.

### Наприклад

Вхідні дані: 10 20 30

Приклад виклику: `python lab3_1.py 10 20 30`

Результат: not triangle

Вхідні дані: 1 1 1

Приклад виклику: `python lab3_1.py 1 1 1`

Результат: triangle

Вхідні дані: 5.5 5.5 -2

Приклад виклику: `python lab3_1.py 5.5 5.5 -2`

Результат: not triangle

2) Програма має розраховувати числа послідовності Фібоначчі.

Послідовність Фібоначчі -- це послідовність чисел, в якій кожний елемент дорівнює сумі двох попередніх. При цьому нульовий елемент вважається за 0, а перший 1. Отже, сама послідовність виглядає наступним чином:  
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, ...

**Вхідні дані:** ціле невід'ємне число n. Передається в програму як аргумент командного рядка.

**Результат роботи:** значення n-го числа послідовності Фібоначчі.

Будь ласка, не використовуйте рекурсію.

### Наприклад

Вхідні дані: 0

Приклад виклику: `python lab3_2.py 0`

Результат: 0

Вхідні дані: 10

Приклад виклику: `python lab3_2.py 10`

Результат: 55

### Завдання №3

1) **Вхідні дані:** рядок, передається в програму як аргумент командного рядка. Може містити пробіли, літери латинського алфавіту в будь-якому регістрі та цифри. Для передачі одним значенням рядок, що містить пробіли, береться в подвійні лапки.

**Результат роботи:** рядок "YES", якщо отриманий рядок є паліндромом, або "NO" - якщо ні. Рядок вважається паліндромом, якщо він однаково читається як зліва направо, так і справа наліво. Відмінністю регістрів та пробілами знехтувати.

### Наприклад

Вхідні дані: 0

Приклад виклику: `python lab4_1.py 0`

Результат: YES

Вхідні дані: puppy

Приклад виклику: `python lab4_1.py puppy`

Результат: NO

Вхідні дані: "mystring1Gni rts ym"

Приклад виклику: python lab4\_1.py "mystring1Gni rts ym"

Результат: YES

2) **Вхідні дані:** довільна, відмінна від нуля, кількість значень - аргументів командного рядка. Значеннями-аргументами можуть бути числа або рядки, які складаються з цифр та літер латинського алфавіту без пробілів.

**Результат роботи:** рядок, що складається з отриманих значень в зворотньому порядку, записаних через пробіл. Пробіл в кінці рядка відсутній.

### Наприклад

Вхідні дані: 1

Приклад виклику: python lab4\_3.py 1

Результат: 1

Вхідні дані: qwe asd zxc 123

Приклад виклику: python lab4\_3.py qwe asd zxc 123

Результат: 123 zxc asd qwe

Вхідні дані: padawan young my HAVE MUST YOU PATIENCE

Приклад виклику: python lab4\_3.py padawan young my HAVE MUST YOU PATIENCE

Результат: PATIENCE YOU MUST HAVE my young padawan

3) **Вхідні дані:** рядок, що складається з відкриваючих та закриваючих круглих дужок - аргумент командного рядка. Для передачі в якості рядка послідовність береться в подвійні лапки.

**Результат роботи:** рядок "YES", якщо вхідний рядок містить правильну дужкову послідовність; або рядок "NO", якщо послідовність є неправильною. Дужкова послідовність вважається правильною, якщо всі дужки можна розбити попарно "відкриваюча"-"закриваюча", при чому в кожній парі закриваюча дужка слідує після відкриваючої.

Пояснення для математиків:

"" (порожній рядок) - правильна дужкова послідовність (ПДП)

"(ПДП)" - також ПДП

"ПДППДП" (дві ПДП записані поряд) - також ПДП

### Наприклад

Вхідні дані: ")"("

Приклад виклику: `python lab4_3.py ")"("`

Результат: NO

Вхідні дані: "(()(())"

Приклад виклику: `python lab4_3.py "(()(())"`

Результат: NO

Вхідні дані: "(()((()()))"

Приклад виклику: `python lab4_3.py "(()((()()))"`

Результат: YES

Вхідні дані: "())()((()())())"

Приклад виклику: `python lab4_3.py "())()((()())())"`

Результат: NO

4) Завдання передбачає пошук "щасливих" квитків. "Щасливим" називається такий тролейбусний квиток, в якому сума перших трьох цифр дорівнює сумі останніх трьох. Наприклад 030111 ( $0+3+0 = 1+1+1$ ), 902326 ( $9+0+2 = 3+2+6$ ), 001100 ( $0+0+1 = 1+0+0$ ).

**Вхідні дані:** два цілих невід'ємних числа ( $0 \leq a1 \leq a2 \leq 999999$ ) - аргументи командного рядка.

**Результат роботи:** кількість "щасливих квитків", чиї номери лежать на проміжку від  $a1$  до  $a2$  включно. Якщо число на проміжку має менше 6 розрядів, вважати, що на його початку дописуються нулі у необхідній кількості, як це і відбувається при нумерації квитків. Виводити номери квитків не треба.

### Наприклад

Вхідні дані: 0 1000

Приклад виклику: `python lab4_4.py 0 1000`

Результат: 1

Пояснення: номер 000000

Вхідні дані: 1001 1122

Приклад виклику: `python lab4_4.py 1001 1122`

Результат: 3

Пояснення: номери 001001, 001010, 001100

Вхідні дані: 222222 222333

Приклад виклику: `python lab4_4.py 222222 222333`

Результат: 7

Пояснения: номера 222222, 222231, 222240, 222303, 222312, 222321, 222330