

Завдання №3

Прикладна аналітика

Задачі на цикли

1. У змінних a і b збережено два цілих додатних числа, таких що $a \leq b$. Напишіть програму, яка знаходить суму всіх чисел від a до b , кратних 3 або 5. Збережіть суму у змінну `result`. Якщо між a і b немає таких чисел, то збережіть у `result` нуль.

Table 1: Приклад

Вхідні дані	Вихідні дані
1, 10	33
9, 9	9
1, 2	0

2. Напишіть програму, яка визначає, чи є задане число `num` простим.

Якщо `num` - просте число, то збережіть у змінну `result` рядок "це просте число".
Якщо число не просте, то збережіть у змінну `result` рядок "це не просте число".

Table 2: Приклад

Вхідні дані	Вихідні дані
7	"це просте число"
8	"це не просте число"

Підказка

Просте число - це число, яке ділиться тільки на 1 і на саме себе.

3. У змінній `number` збережено число. Напишіть код, який перевірить, чи є `number` ступенем двійки? Ряд ступенів двійки:

1, 2, 4, 8, 16, 32, 64 ...

Результат перевірки `True` або `False` збережіть у змінну `is_two_power`.

4. Для приготування однієї порції **еспресо** потрібно приблизно **7 г. меленої кави**. Для приготування **лате** необхідно збити **180 мл. молока** і додати до еспресо, а для приготування **капучино** - **100 мл. молока**.

Задано дві змінні:

- `coffee` - запас меленої кави в кав'ярні (у кілограмах);
- `milk` - запас молока в кав'ярні (у літрах).

Напишіть програму, яка порахує, **скільки відвідувачів** зможе обслужити кав'ярня, якщо відомо, що:

1. У кав'ярні готують три напої на вибір: американо (еспресо + вода), лате і капучино.
2. Кожен відвідувач замовляє тільки один напій.
3. Відомо, що кожен третій замовлений напій - капучино, а кожен п'ятий - лате.
4. Якщо молоко закінчилося, то кав'ярня працює, доки не прийде відвідувач, який бажає замовити каву з молоком.
5. Запас води для американо не обмежений.

Результат збережіть у змінну `visitors`.

Не забудьте перевести кілограми в грами, а літри - в мілілітри.

Table 3: **Приклад**

Вхідні дані	Вихідні дані
0.1, 1	14
0.07, 0.1	4

5. Дано список чисел `a`. Напишіть програму, яка поверне `True`, якщо у списку більше непарних чисел, і `False` - у всіх інших випадках. Результат збережіть у `result`.

Table 4: **Приклад**

Вхідні дані	Вихідні дані
[1, 2, 3, 4, 5, 6, 7, 8, 9]	True
[1, 2, 3, 4, 5, 6, 7, 8, 10]	False

Вхідні дані	Вихідні дані
[1, 2, 3, 4]	False

6. Є словник `grades`, що містить інформацію про шкільні предмети та їхні оцінки. У список `good_subjects` збережіть назви всіх предметів, у яких оцінка більша або дорівнює 7.

Вхідні дані	Вихідні дані
{“Математика”: 9, “Історія”: 6, “Біологія”: 6, “Географія”: 8}	[“Математика”, “Географія”]
{“Алгебра”: 6, “Геометрія”: 6}	[]

7. Напишіть програму, яка прийматиме список чисел `nums` і повертатиме два числа, абсолютна різниця яких мінімальна. Пару чисел збережіть у змінну `result` у вигляді списку, відсортованого за зростанням.

Якщо кілька пар дають мінімальну абсолютну різницю, виберіть ту пару елементів, сума яких більша.

Table 6: **Приклад**

Вхідні дані	Вихідні дані
[40, 16, 8, 17, 15]	[16, 17]
[0, 2, 35, 42, 45, 14, -6, -1]	[-1, 0]

Підказка

- Створіть змінну `min_diff` і збережіть у неї позитивну нескінченність. Це буде змінна зі значенням мінімальної абсолютної різниці. Далі почніть рахувати абсолютну різницю між числами в списку, на кожній ітерації перевіряйте, чи менша поточна різниця, ніж значення у змінній `min_diff`, якщо це так, то зберігайте в `min_diff` поточне значення і продовжуйте пошуки.
 - Запис нескінченності у Python: `float("inf")` або `float("-inf")`.
8. Напишіть програму, що приймає у вигляді аргументів два списки `lst_1` і `lst_2` і визначає, чи є вони протилежними один одному. Результат перевірки збережіть у `result` у форматі `True` або `False`.

Кожна пара списків складатиметься з однакових елементів (типу a і b). Список вважається анти-списком, якщо всі елементи в ньому протилежні відповідним елементам у першому списку.

Вхідні дані	Вихідні дані
["1", "0", "0", "1"], ["0", "1", "1", "0"]	True
["1", "0", "0", "1"], ["0", "1", "1", "1"]	False

9. Задано змінну m - список зі списків однакової довжини. Перевірте, що суми елементів на діагоналях рівні.

Результат перевірки збережіть у result у форматі True або False.

Вхідні дані	Вихідні дані
[[1, 23, 4], [3, 2, 1], [6, 3, 4]]	False
[[1, 23, 4], [3, 2, 1], [1, 3, 4]]	True

10. Напишіть програму, яка приймає словник input_dict і повертає рядок вигляду: ключ=значення&ключ=значення

Рядок має бути лексикографічно відсортований за ключами вихідного словника. Збережіть отриману відповідь у result.

Вхідні дані	Вихідні дані
{ 'lesson': 2, 'task': 21, 'course': 'python' }	'course=python&lesson=2&task=21'

11. Напишіть програму, яка приймає один рядок input_str і повертає інший result, у якому кожна буква вихідного рядка повторюється двічі.

Вхідні дані	Вихідні дані
'python'	'ppyytthhoonn'

12. Напишіть програму, яка прийматиме число n і перевірятиме, чи кратна кожна його цифра цифрі, що стоїть ліворуч від неї. Відповідь збережіть у result у вигляді масиву булевих значень результатів перевірок.

result завжди має починатися з False, оскільки ліворуч від першої цифри нічого немає.

Вхідні дані	Вихідні дані
54412	[False, False, True, False, True]

13. Уявімо, що ми організовуємо іспит, у нас є список студентів, які мають виступити з усною доповіддю. Для виступу відбираються тільки ті студенти, які набрали понад 90 балів за письмову роботу. Наше завдання зробити так, щоб студенти виступали по черзі, відсортовані за прізвищем.

Наприклад, якщо в нас є учасники Бабак, Андющенко, Власенко, то виступати вони мають у такому порядку: Андющенко, Бабак, Власенко.

У словнику `students` збережено дані про студентів, ключ - прізвище, значення - кількість балів за іспит, створіть список `students_order`, у якому кожен елемент - це кортеж (tuple) з номером виступу студента та його прізвищем. У `students_order` мають бути тільки ті студенти, у яких понад 90 балів. Порядок визначається сортуванням за прізвищем.

Вхідні дані	Вихідні дані
{‘Бабак’: 80, ‘Андющенко’: 99, ‘Власенко’: 100}	[(1, ‘Андющенко’), (2, ‘Власенко’)]

14. Давайте навчимося розгадувати таємні шифри! У `my_string` зберігається зашифрований рядок, наприклад:

```
my_string = 'qircxfgwxeciscwgvzlfrciclibecpfvqzg'
```

У словнику `secret_dict` зберігається ключ до шифру, набір символів, які потрібно замінити:

```
secret_dict = {
    'b': 'k',
    'c': 'i',
    'g': 'n',
    'q': 'h',
    'z': 'o',
    'v': 't',
    'w': 'a',
    'f': 'y',
    'x': 'm',
    'r': 'j'}
```

Таким чином, словник `secret_dict` каже нам, що в рядку `my_string` літеру "b" потрібно замінити на "k", "c" на "пробіл" тощо. Якщо ми зробимо такі заміни в рядку `my_string`, то `'qircxfgwxeciscwggwwwzlfrciclibecpfvvqzg'` перетвориться на `'hi, my name is anatoly, i like python'`.

Напишіть програму, яка розшифровує рядок `my_string` за допомогою словника `secret_dict`. Розшифрований рядок збережіть у змінну `decrypted_string`.

15. Дано словник `dict_input`. Поміняйте ключі та значення місцями. Результат збережіть у `result`.

Вхідні дані	Вихідні дані
<code>{"1": 11, "2": 22, "3": 33}</code>	<code>{11: "1", 22: "2", 33: "3"}</code>

Підказка

Варіантів рішення може бути декілька. Один з варіантів - функція `zip()`, яка дозволяє об'єднати дві послідовності у послідовність пар. Або використайте генератор словників.