

## Тема. Типові алгоритми опрацювання даних в одновимірному масиві

### Очікувані результати заняття

Після цього заняття потрібно:

- пояснювати принцип організації даних за допомогою одновимірних масивів.
- називати види опрацювання даних в масиві.

### Поміркуйте

- Якими способами можна заповнити масив даних?
- Які характеристики є у елементів масиву?

### Ознайомтеся з інформацією

#### Основні типи задач опрацювання одновимірних масивів

- **задачі на змінювання значень елементів масиву**  
послідовно перебрати всі елементи масиву і значення чергового елемента змінити за певним правилом;
- **задачі на пошук у масиві елемента із заданою властивістю**  
послідовно перебрати всі елементи масиву; якщо черговий елемент відповідає деякій умові, значення лічильника збільшується на 1;
- **задачі на знаходження суми (добутку) елементів масиву**  
задати початкове значення суми - 0 (добутку - 1), послідовно перебрати всі елементи масиву і додавати (домножувати) значення чергового елемента на поточне значення суми (добутку);
- **задачі на впорядкування (сортування) масивів**  
розстановка елементів масиву в заданому порядку (за зростанням, за зменшенням, за останньою цифрою, в лексикографічному порядку тощо).

### Виконайте вправу

<https://wordwall.net/resource/68179502>

### Робота за комп'ютером

- Завантажте сторінку середовища програмування <https://replit.com/languages/python3> , <https://qaweb.dev/python-online-ua> або у встановленому на смартфоні/планшеті середовищі [Pydroid 3 - IDE for Python 3](#)
- Введіть запропоновані коди до розв'язування задач і запустіть на виконання.
- Визначте, до якого типу задач відноситься дана задача
- Дослідіть принцип роботи кожного коду, змінюючи вхідні умови (кількість елементів, умови відбору елементів тощо)

### Задача 1

Створити список з 10 випадкових елементів (від -10 до 10). Надрукувати лише додатні з них.

#### Розв'язання

Перед тим, як друкувати потрібно перевірити, чи цей елемент є додатним.

1) За допомогою циклу перебираємо всі елементи.

2) Якщо елемент більший нуля, то друкуємо його.

Не забуваємо про відступи! Команди FOR та IF потребують відступів та двокрапок.

### Код

```
import random
spysok=[]
for i in range(10):
    k=random.randint(-10,10)
    spysok.append(k)
print (spysok)

print ("Додатні")
for nomer in range(10):
    if spysok[nomer]>0:
        print (spysok[nomer], end=' ')
```

### Задача 2

Створити список з 10 випадкових елементів (від -10 до 10). Знайти суму цих елементів.

### Розв'язання

Щоб знайти суму елементів списку, потрібно:

- 1) Створити змінну, яка буде накопичувати суму, наприклад s. На початку надати їй значення, s=0
- 2) За допомогою циклу перебираємо усі елементи списку. Можна це робити за номерами чи за самими елементами.
- 3) У циклі додаємо до змінної s кожен елемент, який розглядаємо.
- 4) Після завершення циклу друкуємо результат, тобто значення змінної s.

### Код

```
import random
spysok=[]
for i in range(10):
    k=random.randint(-10,10)
    spysok.append(k)
print (spysok)

s=0
for nomer in range(10):
    s=s+spysok[nomer]
print("Сума=",s)
```

**Виконайте вправи для очей**

### Задача 3

Створити список з 10 випадкових елементів (від -10 до 10). Змінити знак для всіх елементів списку.

### Розв'язання[

- 1) За допомогою циклу перебираємо всі елементи. У цій задачі необхідно перебирати елементи за їх номерами, оскільки потрібно буде змінювати значення цих елементів.
- 2) У циклі змінюємо знак елемента: `spysok[nomer] = - spysok[nomer]`
- 3) Друкуємо список з оновленими значеннями

### Код

```
import random
spysok=[]
for i in range(10):
    k=random.randint(-10,10)
    spysok.append(k)
print (spysok)

print ("Зміна знаку")
for nomer in range(10):
    spysok[nomer]=-spysok[nomer]
print (spysok)
```

### Задача 4

Створити список з 10 випадкових елементів (від -10 до 10). Надрукувати елементи списку у зворотньому порядку.

### Розв'язання

У циклі перебираємо номери елементів списку, але не у прямому порядку (від 0 до 9), а у зворотньому: `range(9,-1,-1)`. Нагадаємо, що в `range` вказуємо початкове значення, кінцеве значення (не включно) та крок зміни. Таким чином, щоб отримати значення від 9 до 0, необхідно починати з 9, завершувати -1 (не включно), і змінювати значення на -1.

Існує функція зміни порядку елементів списку на протилежний: `spysok.reverse()`

### Код

```
import random
spysok=[]
for i in range(10):
    k=random.randint(-10,10)
    spysok.append(k)
print (spysok)

print ("Зворотний порядок")
for nomer in range(9,-1,-1):
    print (spysok[nomer], end=' ')
```

## Домашнє завдання

### Задача 5

Створити список з 10 випадкових елементів (від -10 до 10). Збільшити значення усіх елементів списку удвічі. Знайти суму, кількість та середнє арифметичне окремо додатних та від'ємних елементів списку

У середовищі <https://replit.com/languages/python3> введіть код для розв'язання задачі 5. Зверніть увагу, що дана задача складається з декількох незалежних підзадач - визначте їх. Оцінка за виконану роботу буде залежати від кількості їх виконання. Успіхів!

Скріншоти виконаних кодів надіслати на HUMAN або на електронну пошту [nataliartemiuk.55@gmail.com](mailto:nataliartemiuk.55@gmail.com)

### Джерела

- [Дистосвіта](#)
- [Python – просто!](#)