

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із лабораторної роботи №5
із дисципліни «Основи програмування»
на тему
СТРУКТУРИ ДАНИХ: СПИСКИ,
КОРТЕЖІ, МНОЖИНИ

Виконав:
студент групи КМ-93
Пиндиківський Т. Р.

Керівник:
асистент Дрозденко О. М.

ЗМІСТ

МЕТА РОБОТИ	3
ПОСТАНОВКА ЗАДАЧІ	4
ОСНОВНА ЧАСТИНА	5
ВИСНОВКИ	6
ДОДАТОК А	7
ДОДАТОК Б	10

МЕТА РОБОТИ

Вивчити структури даних (list, tuple, set), створюваних користувачем.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1: Визначити індекси елементів списку, значення яких належать заданому діапазону (тобто не менше заданого мінімуму і не більше заданого максимуму).

Необхідно:

- ☐ заповнити список випадковими числами;
- ☐ запросити для введення мінімум і максимум діапазону і число, з яким треба порівнювати числа з діапазону;
- ☐ знайти індекси елементів, значення яких входять в діапазон. Додавати знайдені індекси до нового списку;
- ☐ вивести загальне число знайдених індексів і окремо всі індекси;
- ☐ елементи списку, які знаходяться згідно з певними індексами, порівняти

з введеним числом. Ті числа, які більше введеного, організувати у список.

Завдання 2:

$$A = \{1, a, ee, ww, c, d\}, B = \{4, a, b, ee, 1, 3\}, F = ((A \oplus B) \cup (A \setminus B))$$

ОСНОВНА ЧАСТИНА

Завдання 1 : Для поетапного виконання програми користувачу спочатку пропонується ввести кількість розмірність списку. Потім список заповнюється довільними числами з використанням функції `random()`. Пізніше, вводяться максимальне та мінімальне значення інтервалу, у рамках значеннях якого будуть шукатися елементи початкового списку. Також, здійснюється пошук елементів списку, значення яких лежать у інтервалі між введеними раніше мінімальним та максимальним елементами. Відбувається виведення самих значень елементів та їх індексів. Також вказується кількість знайдених елементів. Згодом користувачу пропонується ввести числове значення, згідно якого буде здійснено пошук елементів, більших за це значення. Числа, що задовільняють такій умові виводяться у консоль у вигляді списку.

Завдання 2 : Для виконання завдання з операціями над множинами було використано вбудовані в Python функції для виконання обчислень над множинами такі, як : `symmetric_difference()`, `union()`, `difference()`.

ВИСНОВКИ

На цій лабораторній роботі було вивчено структури даних (list, tuple, set), створюваних користувачем. Були розроблені програми, які реалізують поставлену задачу відповідно до вимог. Їх можна застосовувати для вирішення однотипних завдань. Є багато можливостей для розширення функціоналу з метою забезпечення виконання складніших завдань.

ДОДАТОК А

1. Програмна реалізація задачі №1

```
import random
def in_range(element, start_point, finish_point):
    if ((element>=start_point) and (element<=finish_point)):
        return True
    else:
        return False
def start1():
    initial_list=[]
    while True:
        try:
            list_size=int(input("\nВведіть розмір початкового
списку : "))
            break
        except ValueError:
            print("Введіть коректне значення розміру списку")
            continue
    # list_size=random.randrange(100, 1001, 1)
    for i in range (list_size):
        initial_list.append(random.randrange(0, 1001, 1))
    print("\nСписок із елементами :")
    print(initial_list)
    def enter_values():
        while True:
            try:
                start_point=int(input("\nВведіть мінімум
діапазону : "))
                finish_point=int(input("\nВведіть максимум
діапазону : "))
                break
            except ValueError:
                print("Введіть коректні значення мінімуму та
максимуму")
                continue
        return start_point, finish_point
    start_point, finish_point=enter_values()
    # if (((start_point<1) or (start_point>initial_list[list_size-
2]) or (start_point>finish_point)) or ((finish_point<1) or (finish_point>list
_size-1) or (finish_point<start_point))):
    # enter_values()
    if (start_point>finish_point):
        enter_values()
    appropriate_indexes=[]
    for i in range(len(initial_list)):
        if ((initial_list[i]>=start_point) and
(initial_list[i]<=finish_point)):

            appropriate_indexes.append(i)
    appropriate_elements=[initial_list[i] for i in
range(len(initial_list)) if in_range(initial_list[i], start_point,
finish_point)==True]
    print("\nБуло знайдено " + str(len(appropriate_indexes)) + "
елементів, що входять у діапазон : ")
    print(appropriate_elements)
    print("\nЇх індекси : ")
    print(appropriate_indexes)
```

```

while True:
    try:
        number_to_compare=int(input("\nВведіть значення числа
для порівняння : "))
        break
    except ValueError:
        print("Введіть коректне значення числа для порівняння")
        continue
    appropriate_elements=[element for element in (initial_list) if
element>number_to_compare]
    appropriate_indexes=[i for i in range(len(initial_list)) if
initial_list[i]>number_to_compare]
    print("\nЕлементи, які більші вказаного значення : ")
    print(appropriate_elements)
    print("\nЇх індекси : ")
    print(appropriate_indexes)

#start1()

```

```

C:\Users\taras\Desktop\#5 (master -> origin)
λ python menu.py

Лабораторна робота №4
Пиндиківський Тарас, КМ-93
16 варіант

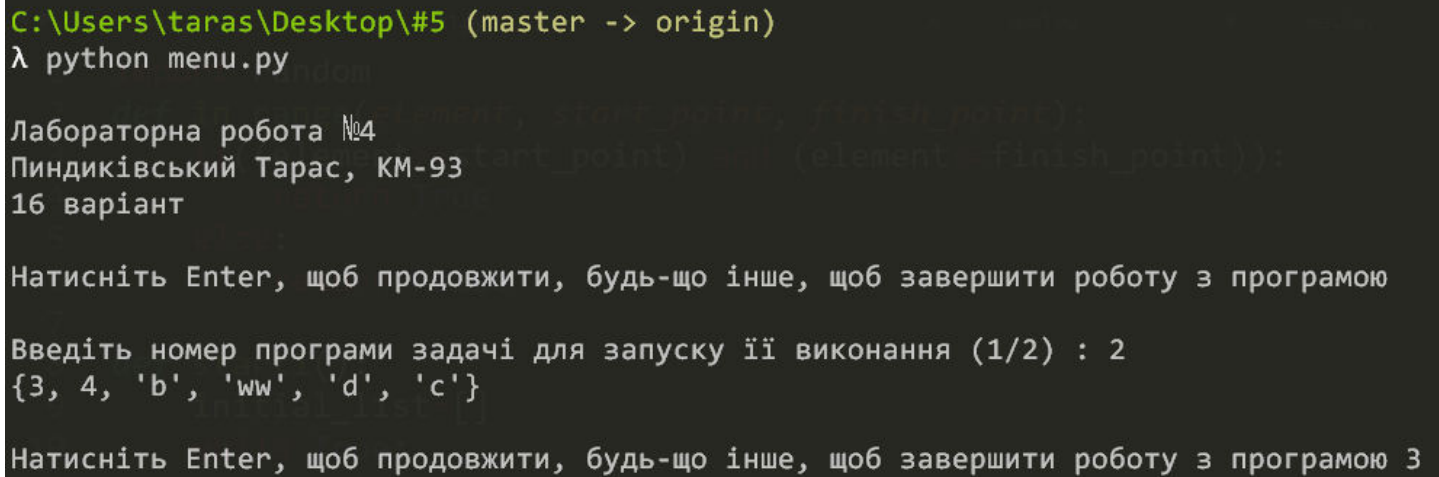
Натисніть Enter, щоб продовжити, будь-що інше, щоб завершити роботу з програмою
Введіть номер програми задачі для запуску її виконання (1/2) : 1
Введіть розмір початкового списку : 10
Список із елементами :
[372, 764, 505, 292, 758, 365, 587, 149, 841, 276]
Введіть мінімум діапазону : 225
Введіть максимум діапазону : 784
Було знайдено 8 елементів, що входять у діапазон :
[372, 764, 505, 292, 758, 365, 587, 276]
Їх індекси :
[0, 1, 2, 3, 4, 5, 6, 9]
Введіть значення числа для порівняння : 489
Елементи, які більші вказаного значення :
[764, 505, 758, 587, 841]
Їх індекси :
[1, 2, 4, 6, 8]
Натисніть Enter, щоб продовжити, будь-що інше, щоб завершити роботу з програмою 3

```

Рис. 1 - програмна реалізація задачі №1

2. Програмна реалізація задачі №2

```
def start2():  
    a={1, "a", "ee", "ww", "c", "d"}  
    b={4, "a", "b", "ee", 1, 3}  
    f=(a.symmetric_difference(b)).union(a.difference(b))  
    print(f)
```



```
C:\Users\taras\Desktop\#5 (master -> origin)  
λ python menu.py  
Лабораторна робота №4  
Пиндиківський Тарас, КМ-93  
16 варіант  
  
Натисніть Enter, щоб продовжити, будь-що інше, щоб завершити роботу з програмою  
Введіть номер програми задачі для запуску її виконання (1/2) : 2  
{3, 4, 'b', 'ww', 'd', 'c'}  
Натисніть Enter, щоб продовжити, будь-що інше, щоб завершити роботу з програмою 3
```

Рис. 2 - програмна реалізація задачі №2

ДОДАТОК Б

Питання для самоперевірки

1. Що таке список, кортеж, множина.

Списки - найбільш багатofункціональний тип даних, який записується як список елементів, розділених комами, і укладених в квадратні дужки. Кортеж - це незмінний список. Кортеж не може бути змінений ніяким способом після його створення.

Множина в мові Python - це структура даних, еквівалентна множинам в математиці. Множина може складатися з різних елементів. Цей тип даних є нерегульованим, як наслідок неіндексованим, змінним і ітеративним.

2. Перерахуйте відмінності списку від кортежу.

1. Кортеж визначається так само, як список, за винятком того, що набір елементів укладається в круглі дужки, а не в квадратні.

2. Елементи кортежу задані в певному порядку, як і в списку. Елементи кортежу індексуються з нуля, як і елементи списку, таким чином, перший елемент не порожнього кортежу - це завжди `a_tuple[0]`.

3. Від'ємні значення індексу відраховуються від кінця кортежу, як і в списку. Останній елемент має індекс -1.

4. Створення зрізу кортежу («slicing») аналогічно створенню зрізу списку. Коли створюється зріз списку, виходить новий список; коли створюється зріз кортежу, виходить новий кортеж.

5. Основна відмінність між кортежами і списками полягає в тому, що кортежі не можуть бути змінені. На практиці це означає, що у них немає методів, які б дозволили їх змінити. У списках є такі методи, як `append()`, `extend()`, `insert()`, `remove()`, `pop()`. У кортежів жодного із цих методів немає.

3. Наведіть приклади створення списку, кортежу, множини.

Список можна створити за допомогою оператора `[]` або методу `list`.

`a_tuple=("a","b","z","example")` # створюємо кортеж

4. Перерахуйте методи зміни списку.

`weekdays = ['Monday', 'Tuesday', 'Wednesday']`

5. Перерахуйте функції вищого порядку для роботи з послідовностями.

`Map()`, `reduce()`, `filter()`, `zip`.

6. Що повертає функція `zip`.

Функція `zip` на кожній ітерації повертає кортеж, що містить елементи послідовностей, які розташовані на однаковому зміщенні.

7. Як кортеж перетворити у список.

Функцією `list()`.

8. Скільки разів елемент може входити до множини.

Тільки один раз.

9. Чи може список бути елементом множини.

Не можна зробити елементом множини список (але можна зробити кортеж).

10. Як можна із множини зробити список.

Функцією `list()`.

11. Чи можна у Python виконувати звичайні для математики операції над множинами.

Так – перетин, об'єднання, симетрична різниця...