НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №5

iз дисципліни «Програмування»

на тему

«СТРУКТУРИ ДАНИХ: СПИСКИ,КОРТЕЖИ, МНОЖИНИ»

Виконав: Керівник:

студент групи КМ-83 ст.вик.

Касіч Б.В. Дрозденко О.М.

Київ — 2018

1. ПОСТАНОВКА ЗАДАЧІ

1.1 Мета роботи

Вивчення структур даних (list, tuple, set), створюваних користувачем.

1.2 Що потрібно знати

- що таке список, кортеж, множина;

- операції зі списками;

- перетворення списку в інші типи даних.

1.3 Узагальнене формулювання завдання до лабораторної роботи

1) Вивчити принципи створення списків і множин у мові Python.

2) Розробити програми відповідно до варіанта завдання.

3) Вхідні дані і результат роботи супроводжувати відповідною

інформацією на екрані.

4) Показати розроблену програму викладачеві.

5) Письмово відповісти на Питання для самоперевірки.

6) Оформити звіт відповідно до вимог.

Завдання на виконання лабораторної роботи складається з 2-х частин:

- робота зі списками;

- робота з множинами.

2. ОПИС ПРОГРАМ

2.1 Перша програма

За допомогою команди print ми вітаємо користувача та надаємо йому інформацію про дану програму. Визначаємо рекурсивну функцію common\_elems(a,b) (а – перший список, b – другий список) , яка в своєму тілі здатна до порожнього списку common додати усі загальні елементи введених користувачем, відсортованих списків, а потім повернути цей список. Це зроблено за допомогою чотирьох умовних методів if та рекурсивного повернення функції ,із зменшенням відповідних списків, залежно від умови, до моменту зпустощення одного із двох списків. В кінці повертається список common.

Визначаємо функцію get\_list(a) (a – номер списку), яка дозволяє отримати від користувача список цілих чисел, тобто функція задля валідації отриманих данних. Зроблено це за допомогою методу try-except, тобто при введені не цілого числа функцію буде рекурсивно повернено із командою print("Вводьте коректні дані, будь ласка"), замість виведення помилки. При вдалому введені повертається попередньо згенерований список за допомогою методу генерування

[int(i) for i in input().split()].

Далі створюється порожній список common. До двох змінних list\_a та list\_b присвоюється функція get\_list(“номер списку”). Потім списки відсортовуються в порядку зростання за допомогою методу sort(). І потім безпосередьно виводиться за допомогою команд print перший список , другий , та функція common\_elems(list\_a,list\_b) , що поверне нам список загальних елементів даних списків.

Далі за допомогою команди inputми запитуємо у користувача ,

хоче він завершити програму чи ні. Якщо хоче , то він має написати

stopі програма завершиться не продовживши цикл while, який ми раніше створили після визначення функцій. В цьому циклі безпосередьно знаходиться увесь код без визначення функцій.

2.2 Друга програма

За допомогою команди print ми вітаємо користувача та надаємо йому інформацію про дану програму. Імпортуємо бібліотеку itertools(бібліотека ітерацій). Безпосередьно визначаємо дані нам множини A і B , даний універсум Univers (тобто присвоюємо значення змінним). Потім присвоюємо змінним відповідні операції над множинами ( об’єднання, перетин і т.д.). Це зроблено таким способом: var=set1.some\_opperation(set2).

За допомогою команд print виводимо наші списки та безпосередьно результат операцій над ними. Для виведення декартових добутків використовуємо модуль product із бібліотеки itertools, що створить відповідні пари наших множин і додасть їх у порожні множини, які ми створили раніше, за допомогою методу set.add(elem). Потім безпосередньо виводимо ці списки на екран за допомогою команди print. Програма завершиться після того , як користувач натисне Enter.

Дані множини:



Умова особистого завдання:



3. РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

3.1 Усі можливі результати першої програми продемонстровано на

Рисунку 3.1

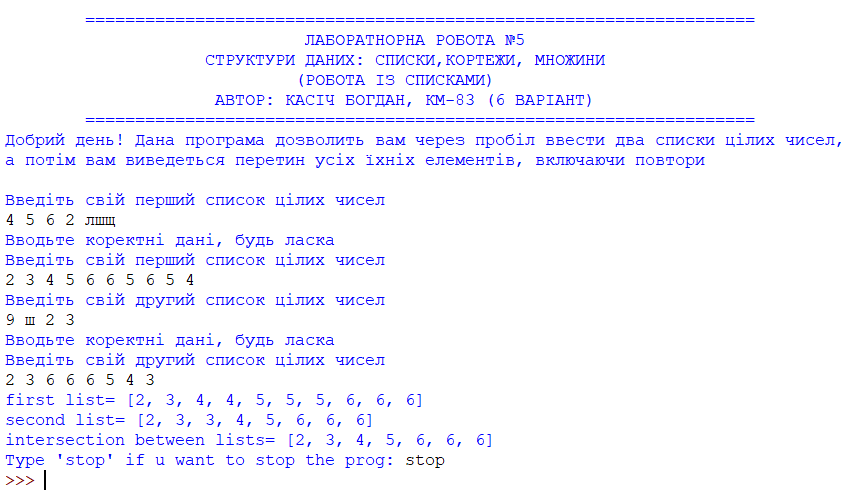


Рисунок 3.1

3.2 Усі можливі результати другої програми продемонстровано на

Рисунку 3.2

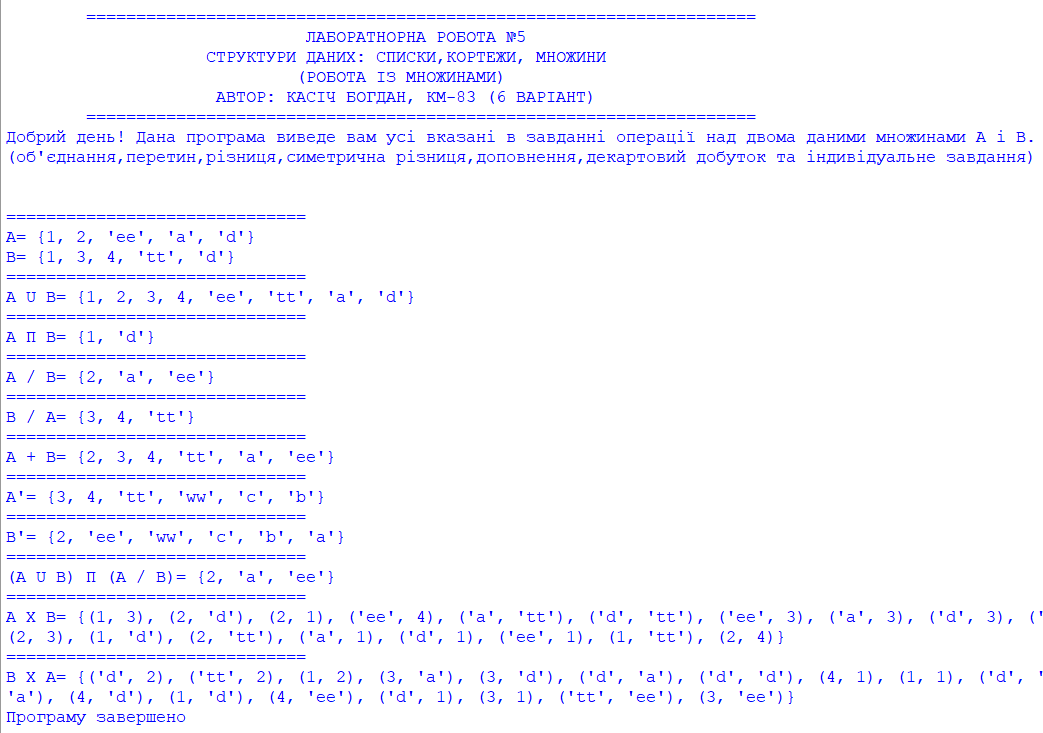
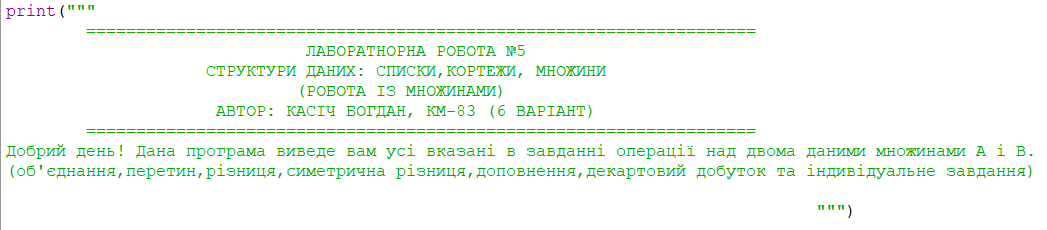


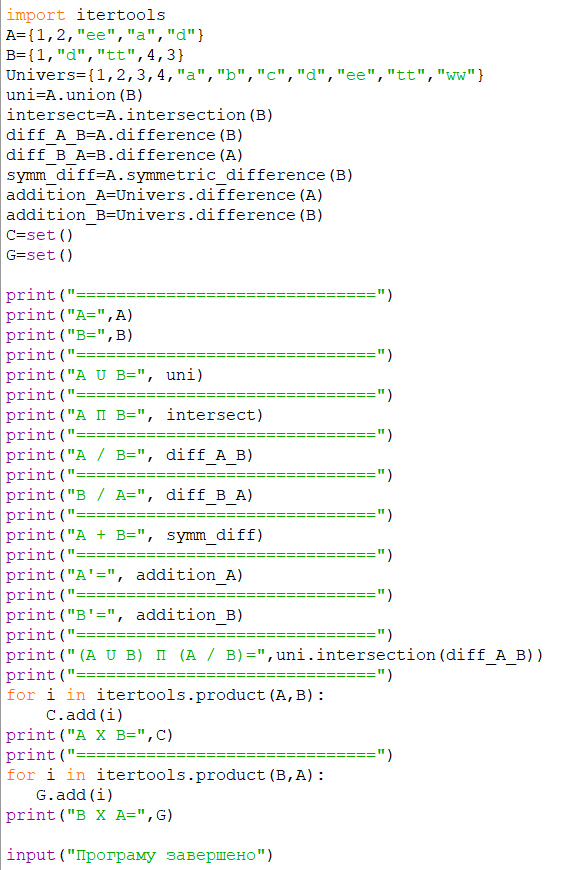
Рисунок 3.2

Текст першої програми:



Текст другої програми:





Відповіді до контрольних запитань

1. Що таке список, кортеж, множина.

Список – це послідовність елементів, пронумерованих від 0

Кортеж – це послідовність незмінних об'єктів

Множина – набір або сукупність об'єктів

2. Перерахуйте відмінності списку від кортежу.

В кортежі не можна змінювати елементи та кортеж ініціалізується у круглих дужках, коли список у квадратних

3. Наведіть приклади створення списку, кортежу, множини.

a=[] – list

b=set() – set

c=() – tuple

4. Перерахуйте методи зміни списку.

append,remove,extend,insert,reverse,sort,list[index]=”smth”…

5. Перерахуйте функції вищого порядку для роботи з послідовностями.

max,min,len…

6. Що повертає функція *zip*.

Zip створює об'єкт-ітератор, з якого при кожному оберті циклу витягується кортеж, що складається з двох частин.

7. Як кортеж перетворити у список.

some\_tuple=list(some\_tuple)

8. Скільки разів елемент може входити до множини.

Один або нуль

9. Чи може список бути елементом множини.

Так

10. Як можна із множини зробити список.

some\_set=list(some\_set)

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

множинами.

Так

ЗМІСТ

1.ПОСТАНОВКА ЗАДАЧІ…………………….…………………………………2

2.ОПИС ПРОГРАМ………………………………………………………….…3-4

Результати випробувань……………………………………………………….5-6

Тексти програм…………………………………………………………………7-8

Відповіді до контрольних запитань……………………………………………9