**Практичні завдання**

Напишіть програму на Python, яка :

1. Сортує словник за значеннями
2. Додає ключ до словника : Було : {0: 10, 1: 20} Результат : {0: 10, 1: 20, 2: 30}
3. Поєднає наступні словники та створює новий

dic1={1:10, 2:20}

dic2={3:30, 4:40}

dic3={5:50,6:60}

Результат : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

1. Перевіряє чи входить даний ключ до словника
2. Ітерується по словнику використовуючи for loops.
3. Генерує та виводить словник, який складається с чисел від 1 до n в формі (x, x\*x). Ввід ( n = 5). Результат : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
4. Генерує словник у якому ключі це числа від 1 до 15, а значення квадрати ключів. Приклад {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}
5. Яка створює будь який set.
6. Яка ітерується крізь sets.
7. Додає значення до set.
8. Видаляє елемент з set
9. Видаляє елемент якщо він є в set.
10. Створює об’єднання sets (intersection).
11. Яка знаходить суму всіх елементів списку.
12. Яка помножує всі елементи списку
13. Яка виводить найбільше та найменше значення у списку.

17. Напишіть програму Python, щоб отримати список, відсортований у порядку зростання за останнім елементом у кожному кортежі із заданого списку не порожніх кортежів.

Приклад : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)] Результат : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]

18. Яка видаляє елементі які повторюються

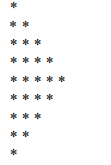
19. Яка друкує лист після видалення 0th, 4th та 5th елементів. Приклад : ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow'] Результат : ['Green', 'White', 'Black']

20. Програмка яка помножує всі значення в списку. (8, 2, 3, -1, 7) Результат: -336

21. Розраховує факторіал числа.

22. Приймає строку та підраховує кількість upper case letters and lower case letters. Ввід : 'The quick Brow Fox' Результат : No. of Upper case characters : 3 No. of Lower case Characters : 12

23. Друкує наступний патерн, використовуючи цикли



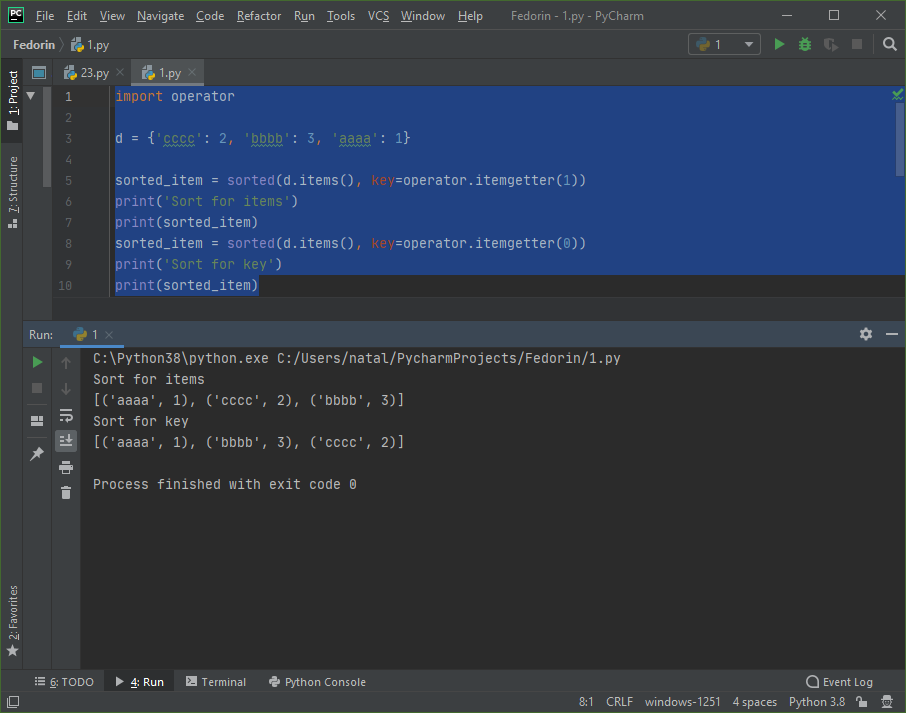
**Результати роботи програмного коду**

1) Сортує словник за значеннями

**Програмна реалізація:**

import operator  
  
d = {'cccc': 2, 'bbbb': 3, 'aaaa': 1}  
  
sorted\_item = sorted(d.items(), key=operator.itemgetter(1))  
print('Sort for items')  
print(sorted\_item)  
sorted\_item = sorted(d.items(), key=operator.itemgetter(0))  
print('Sort for key')  
print(sorted\_item)

**Результат:**

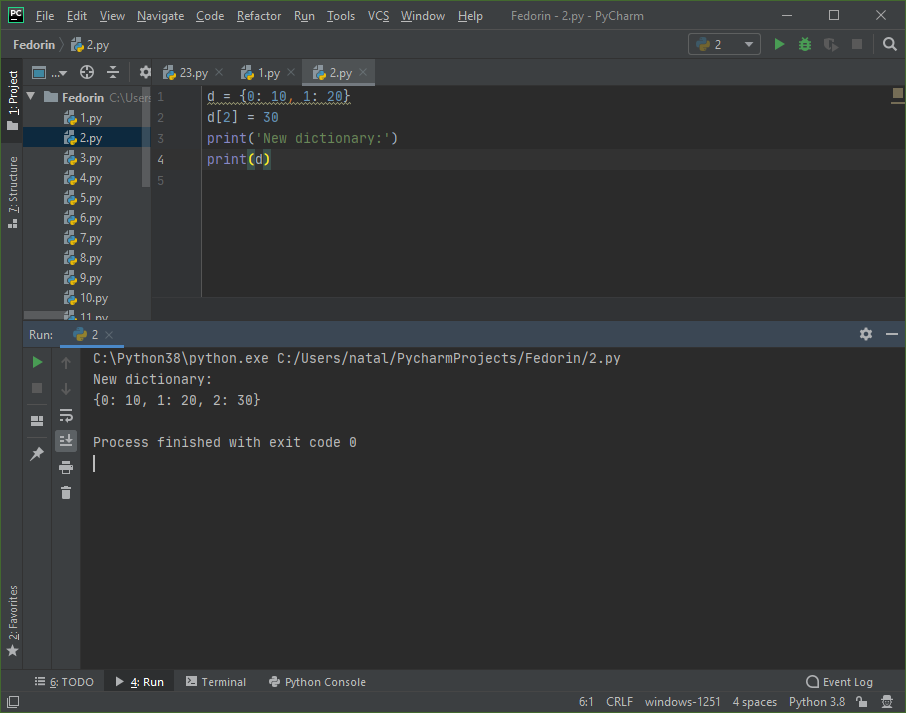


2) Додає ключ до словника

**Програмна реалізація:**

d = {0: 10, 1: 20}  
d[2] = 30  
print('New dictionary:')  
print(d)

**Результат:**

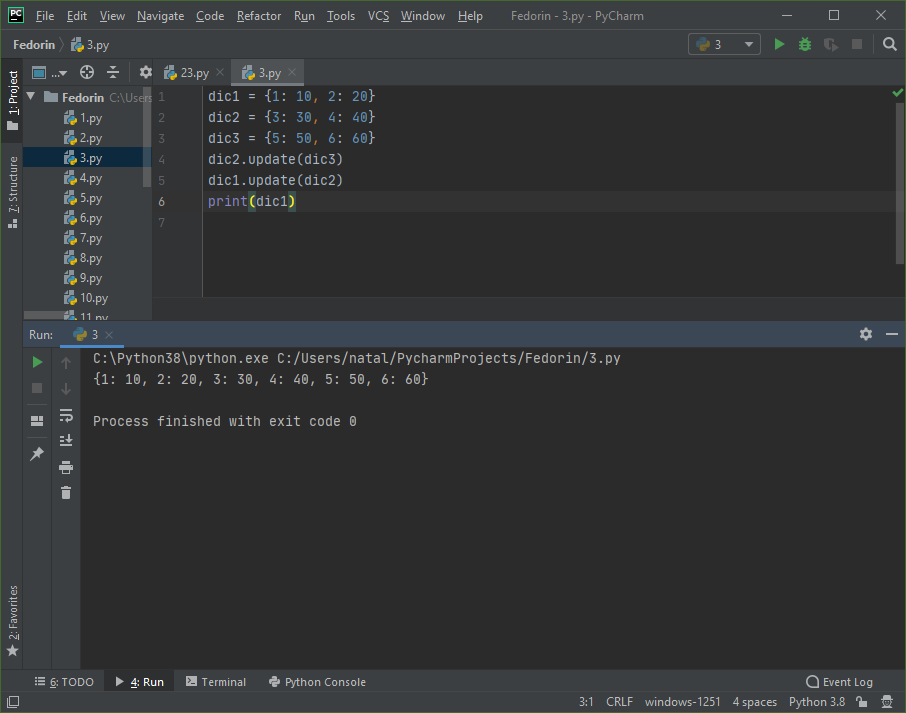


3) Поєднає наступні словники та створює новий

**Програмна реалізація:**

dic1 = {1: 10, 2: 20}  
dic2 = {3: 30, 4: 40}  
dic3 = {5: 50, 6: 60}  
dic2.update(dic3)  
dic1.update(dic2)  
print(dic1)

**Результат:**

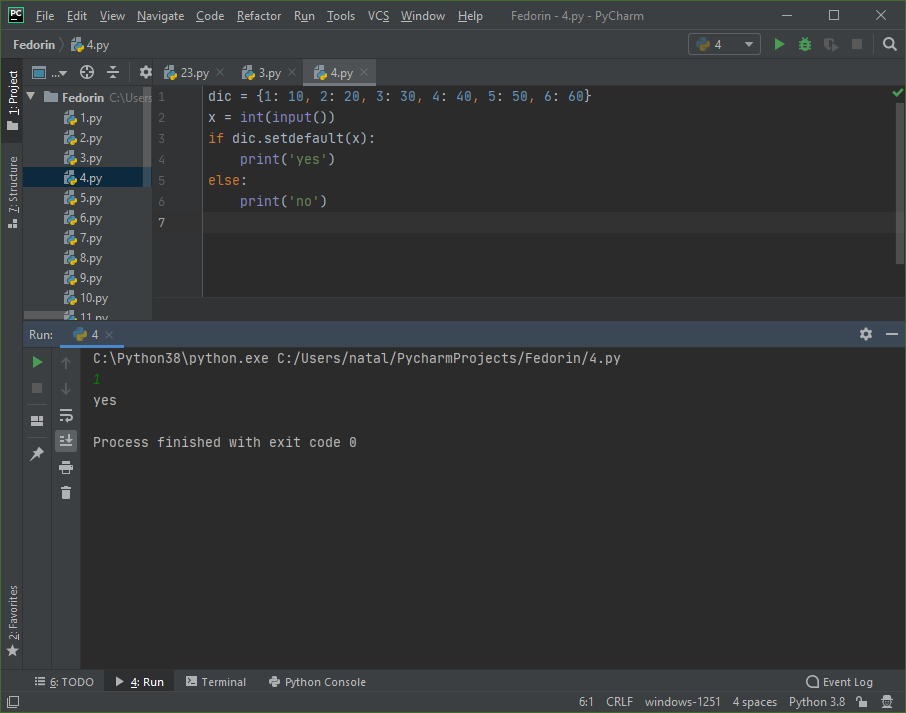


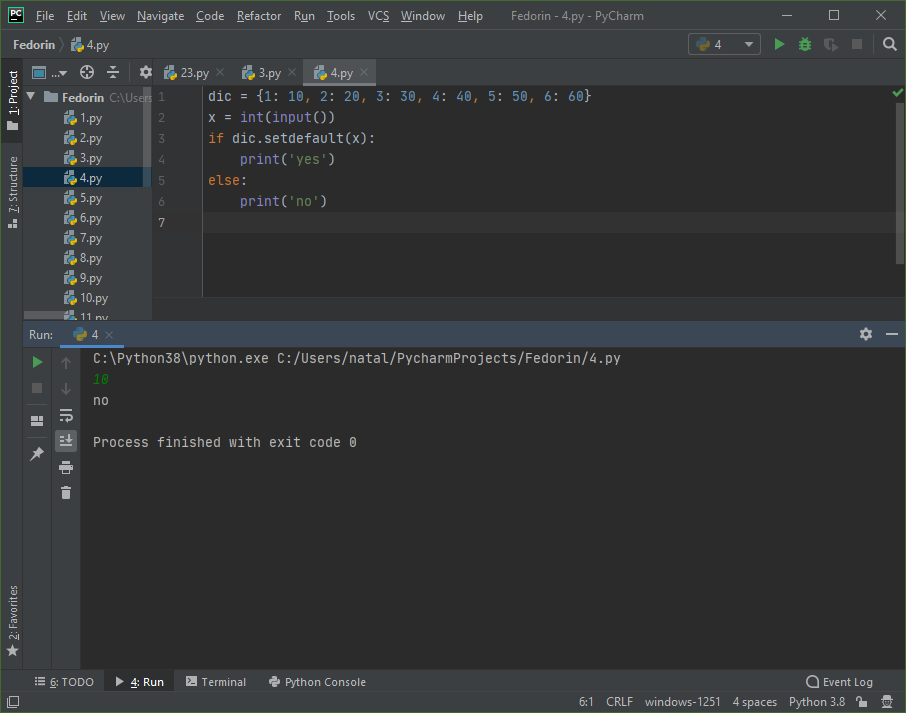
4) Перевіряє чи входить даний ключ до словника

**Програмна реалізація:**

dic = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}  
x = int(input())  
if dic.setdefault(x):  
 print('yes')  
else:  
 print('no')

**Результат:**



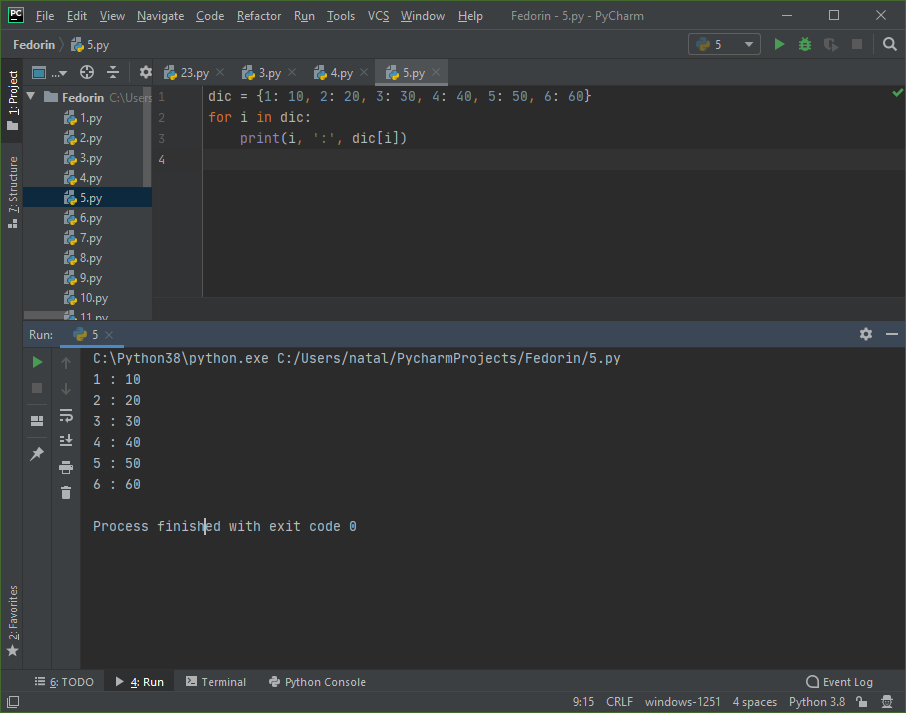


5) Ітерується по словнику використовуючи for loops

**Програмна реалізація:**

dic = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}  
for i in dic:  
 print(i, ':', dic[i])

**Результат:**

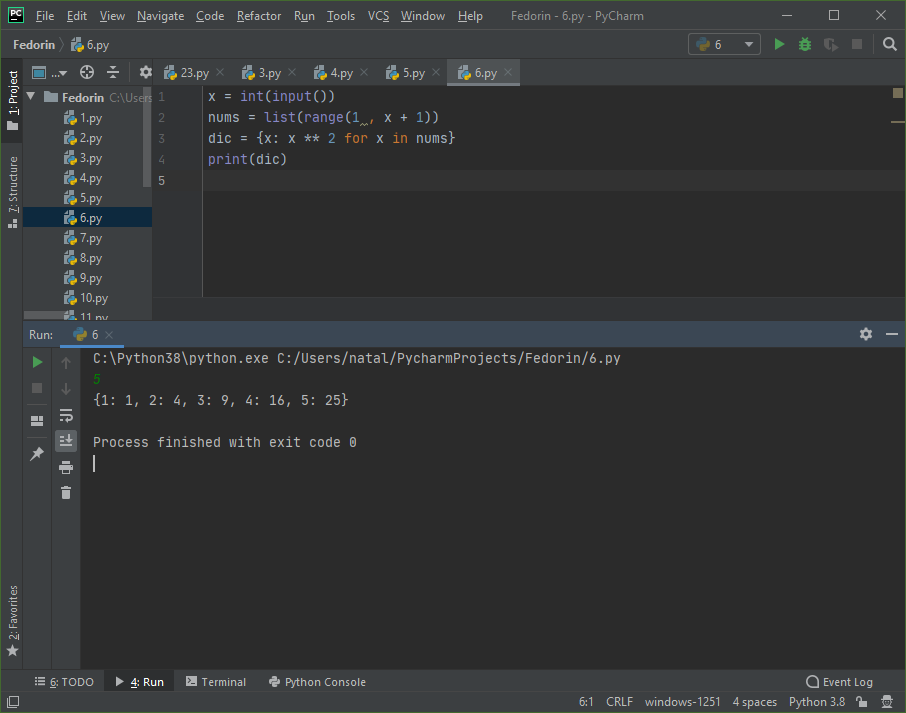


6) Генерує та виводить словник, який складається с чисел від 1 до n в формі (x, x\*x)

**Програмна реалізація:**

x = int(input())  
nums = list(range(1 , x + 1))  
dic = {x: x \*\* 2 for x in nums}  
print(dic)

**Результат:**

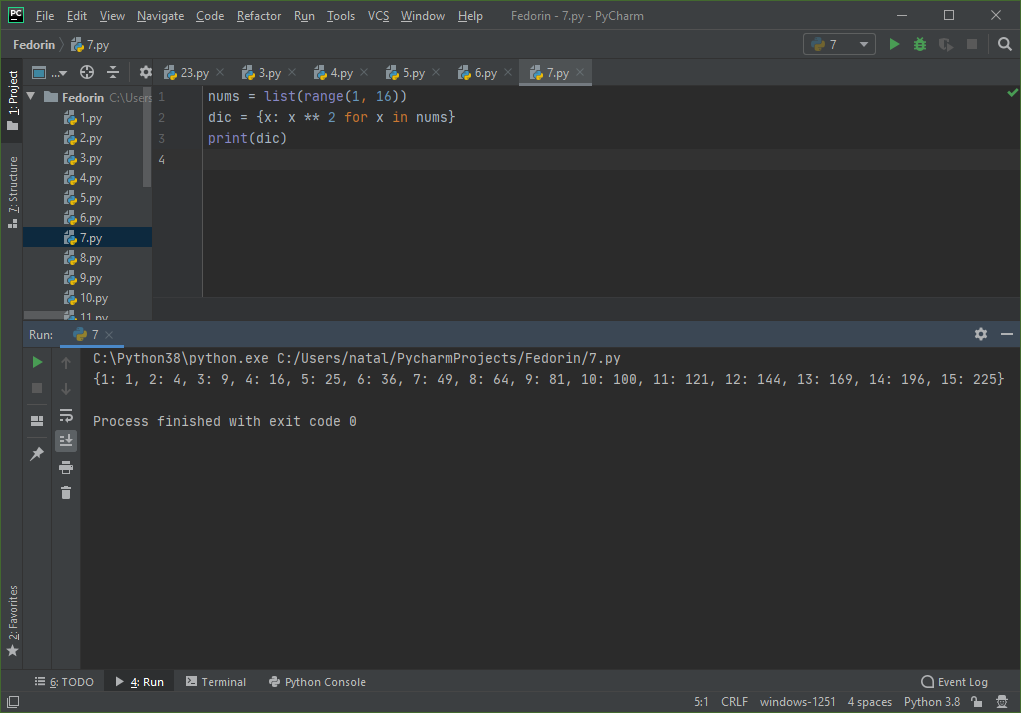


7) Генерує словник у якому ключі це числа від 1 до 15, а значення квадрати ключів

**Програмна реалізація:**

nums = list(range(1, 16))  
dic = {x: x \*\* 2 for x in nums}  
print(dic)

**Результат:**

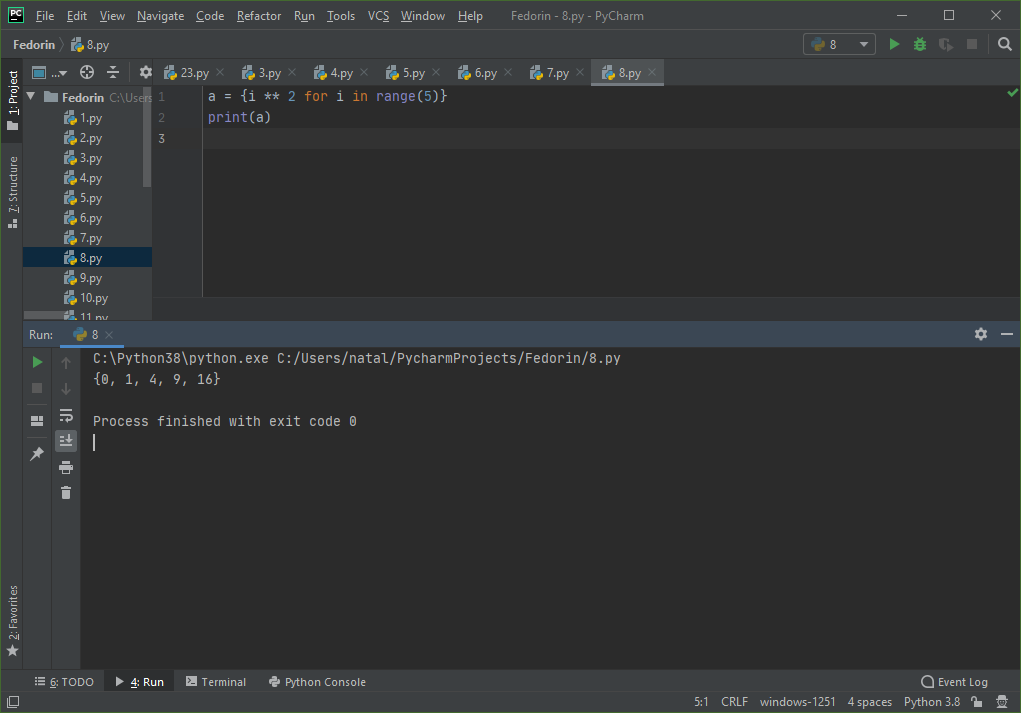


8) Яка створює будь який set.

**Програмна реалізація:**

a = {i \*\* 2 for i in range(5)}  
print(a)

**Результат:**

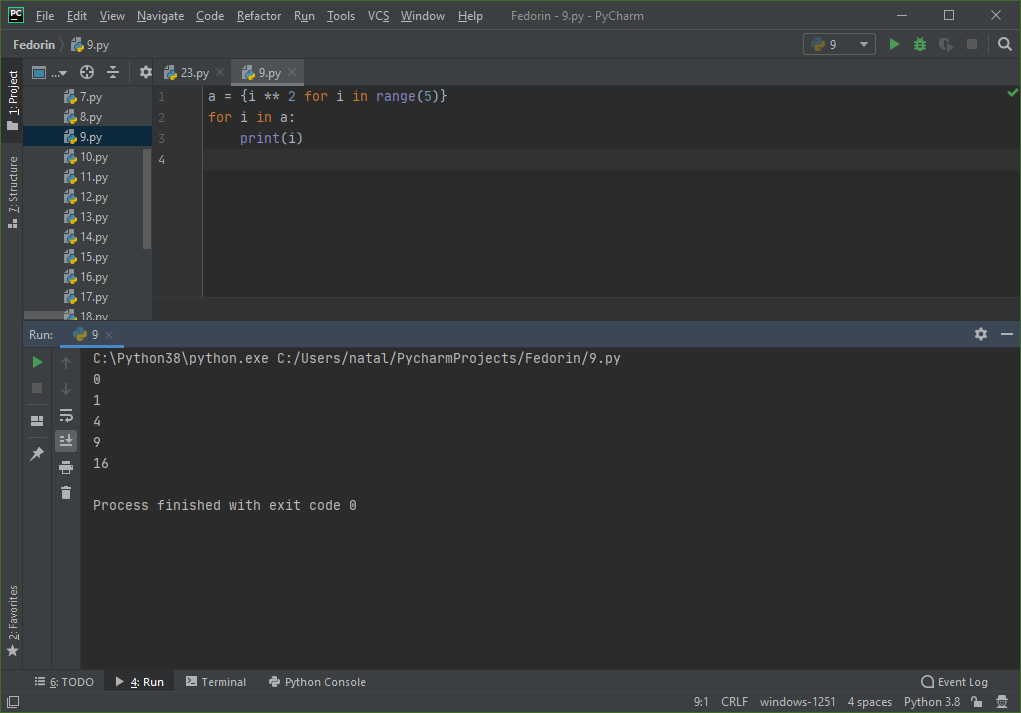


9) Яка ітерується крізь sets

**Програмна реалізація:**

a = {i \*\* 2 for i in range(5)}  
for i in a:  
 print(i)

**Результат:**

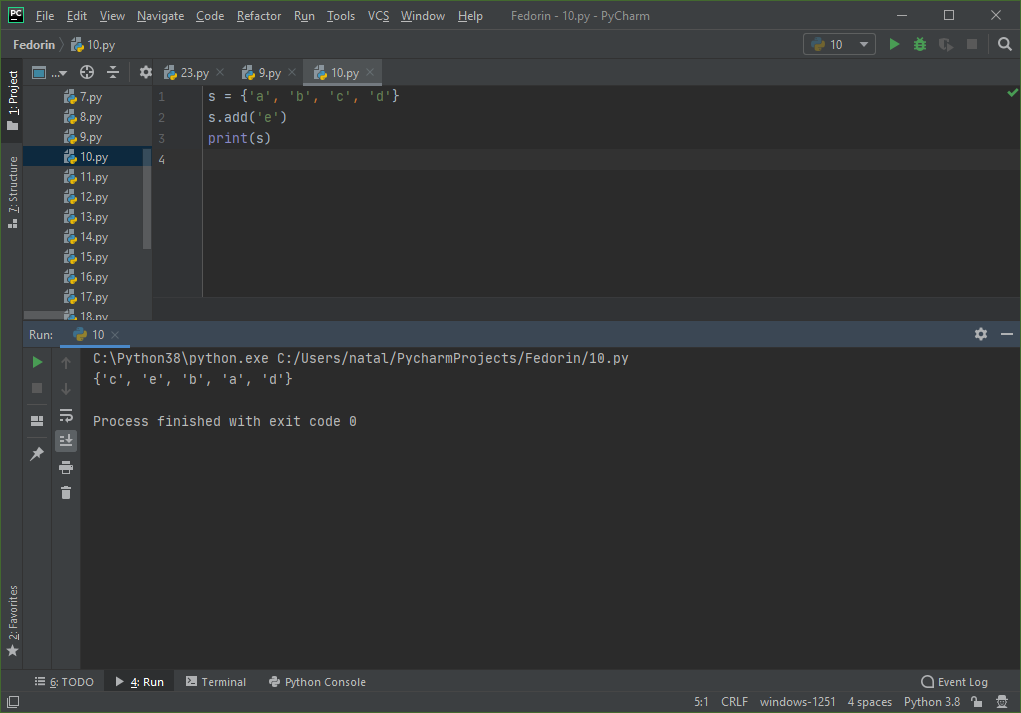


10) Додає значення до set

**Програмна реалізація:**

#Додає значення до set.  
s = {'a', 'b', 'c', 'd'}  
s.add('e')  
print(s)

**Результат:**

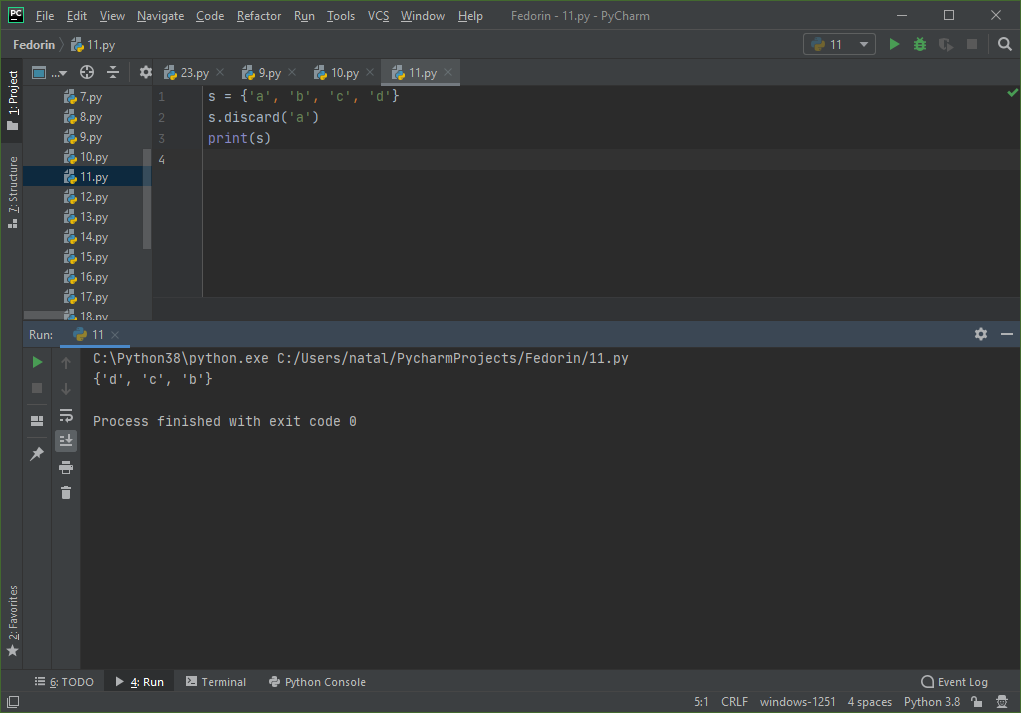


11) Видаляє елемент з set

**Програмна реалізація:**

s = {'a', 'b', 'c', 'd'}  
s.discard('a')  
print(s)

**Результат:**

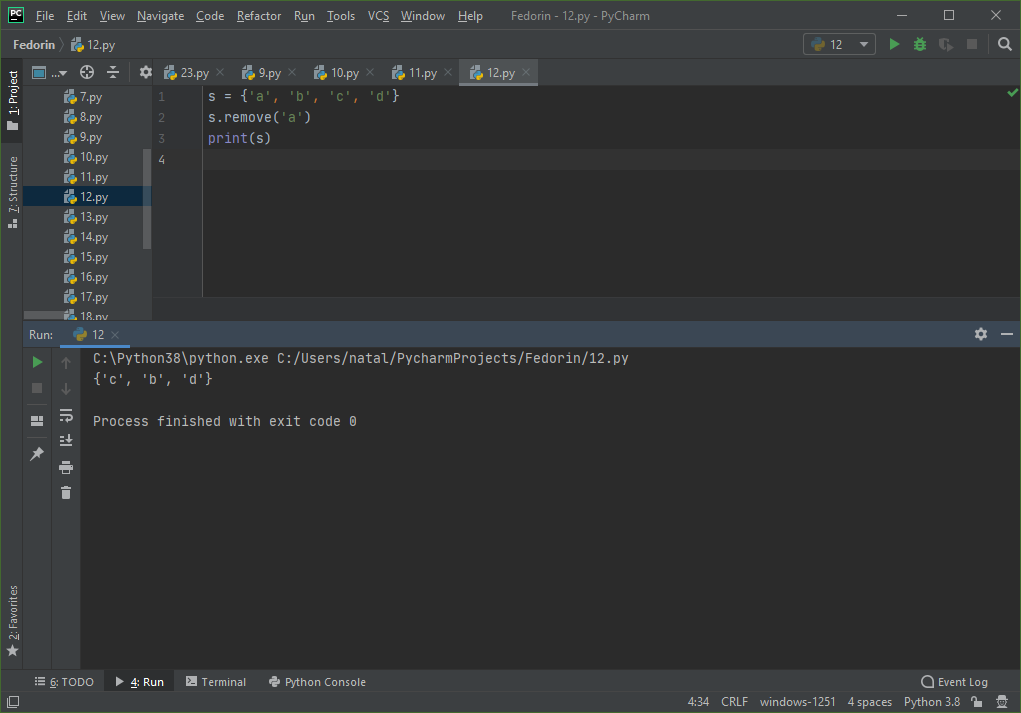


12) Видаляє елемент якщо він є в set

**Програмна реалізація:**

s = {'a', 'b', 'c', 'd'}  
s.remove('a')  
print(s)

**Результат:**

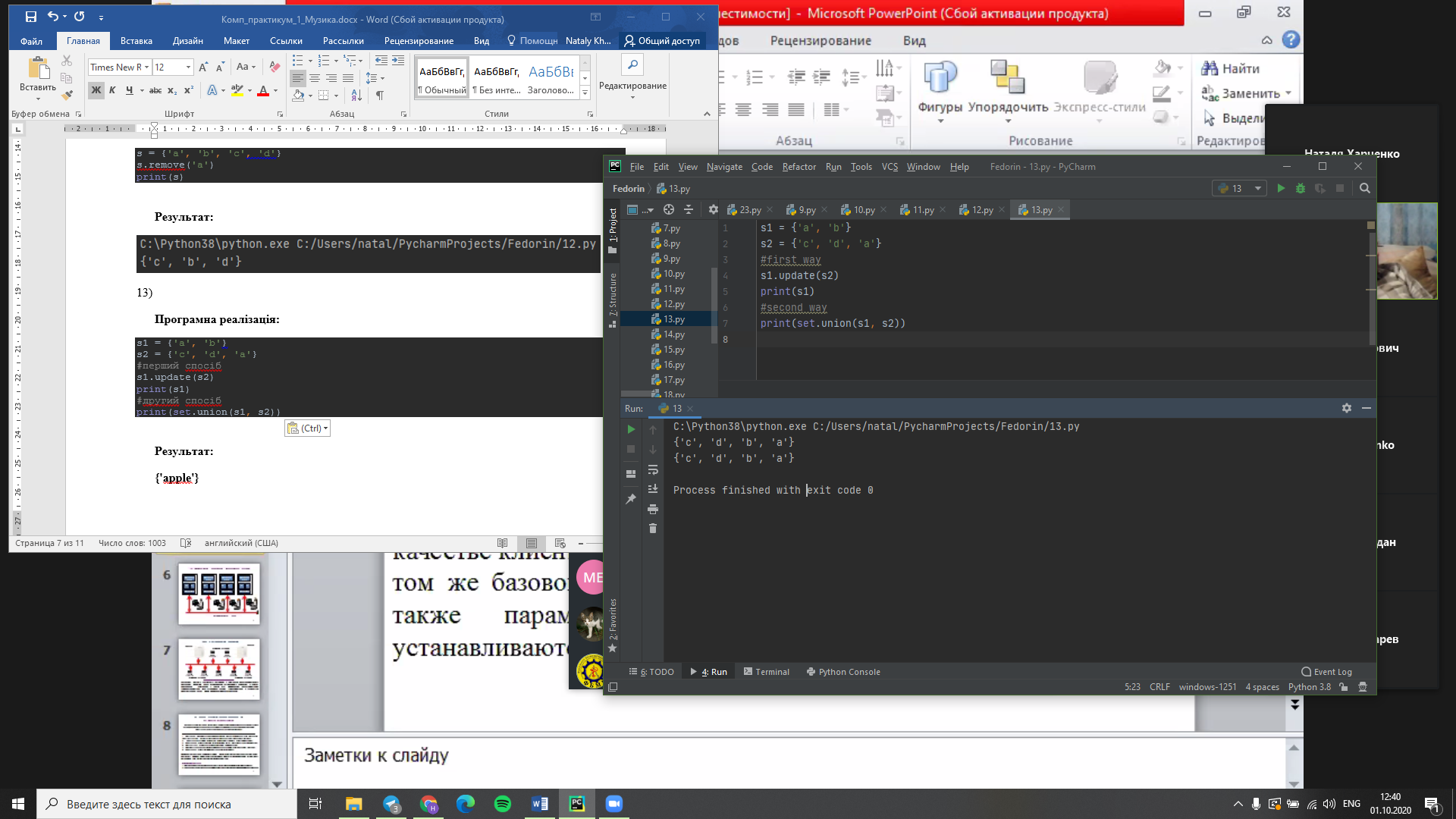


13) Створює об’єднання sets (intersection).

**Програмна реалізація:**

s1 = {'a', 'b'}  
s2 = {'c', 'd', 'a'}  
#перший спосіб  
s1.update(s2)  
print(s1)  
#другий спосіб  
print(set.union(s1, s2))

**Результат:**

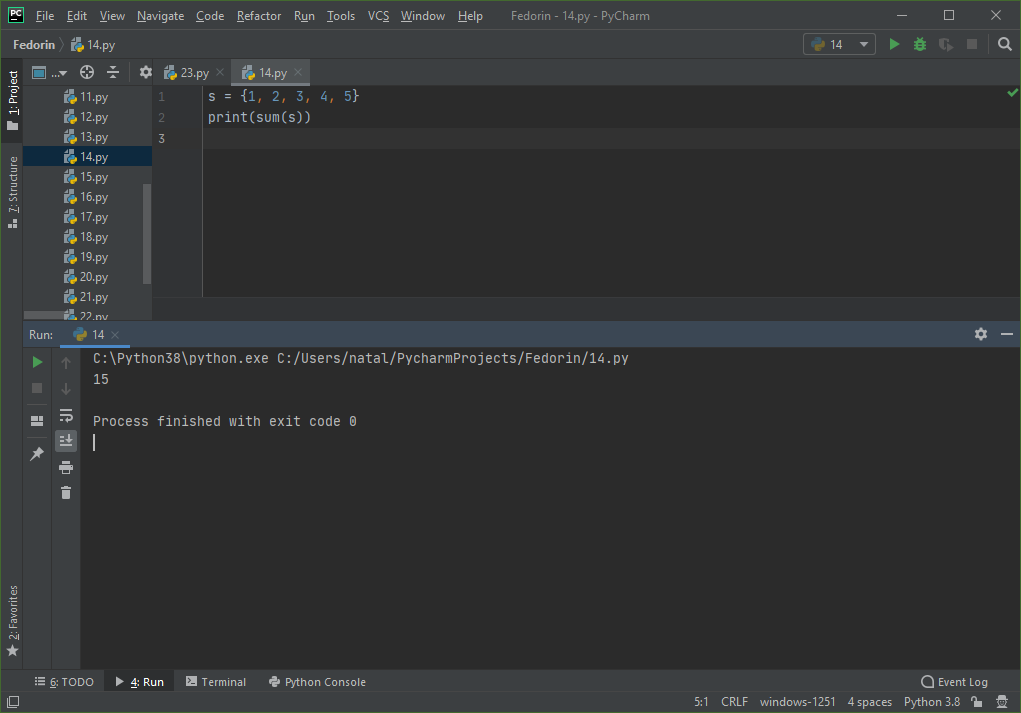


14) Яка знаходить суму всіх елементів списку

**Програмна реалізація:**

s = {1, 2, 3, 4, 5}  
print(sum(s))

**Результат:**

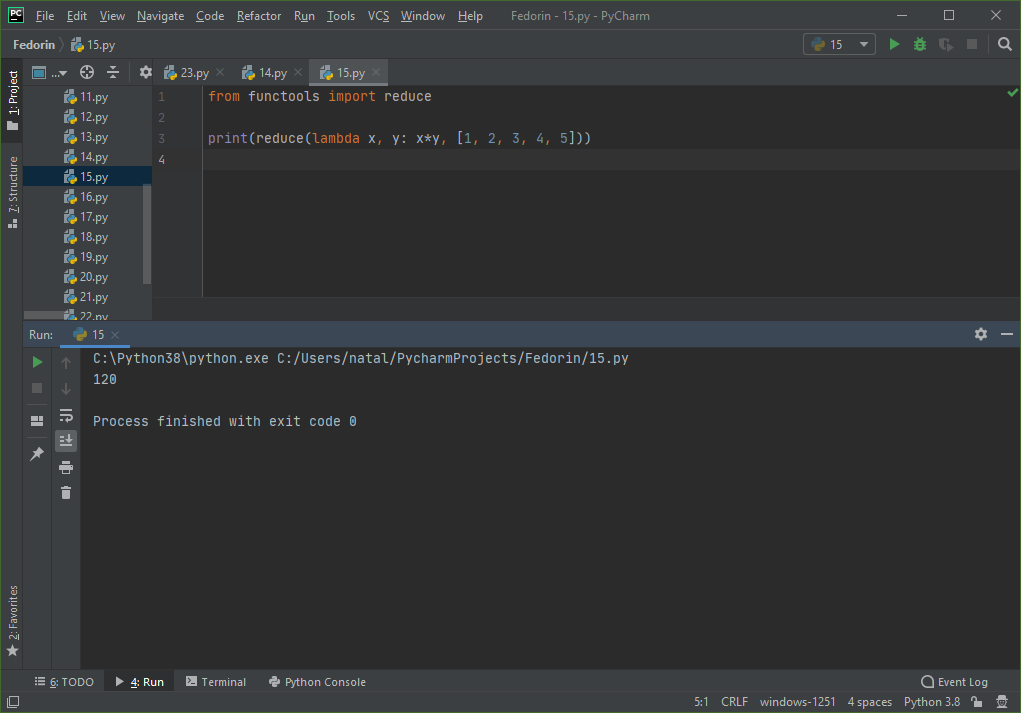


15) Яка помножує всі елементи списку

**Програмна реалізація:**

from functools import reduce  
  
print(reduce(lambda x, y: x\*y, [1, 2, 3, 4, 5]))

**Результат:**

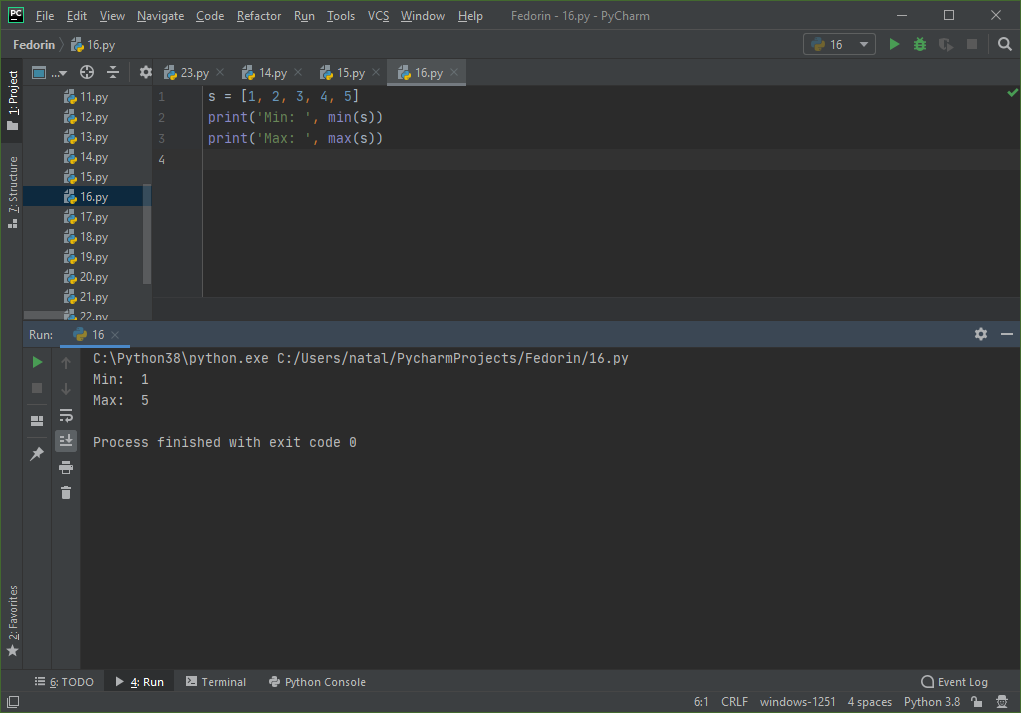


16) Яка виводить найбільше та найменше значення у списку

**Програмна реалізація:**

s = [1, 2, 3, 4, 5]  
print('Min: ', min(s))  
print('Max: ', max(s))

**Результат:**

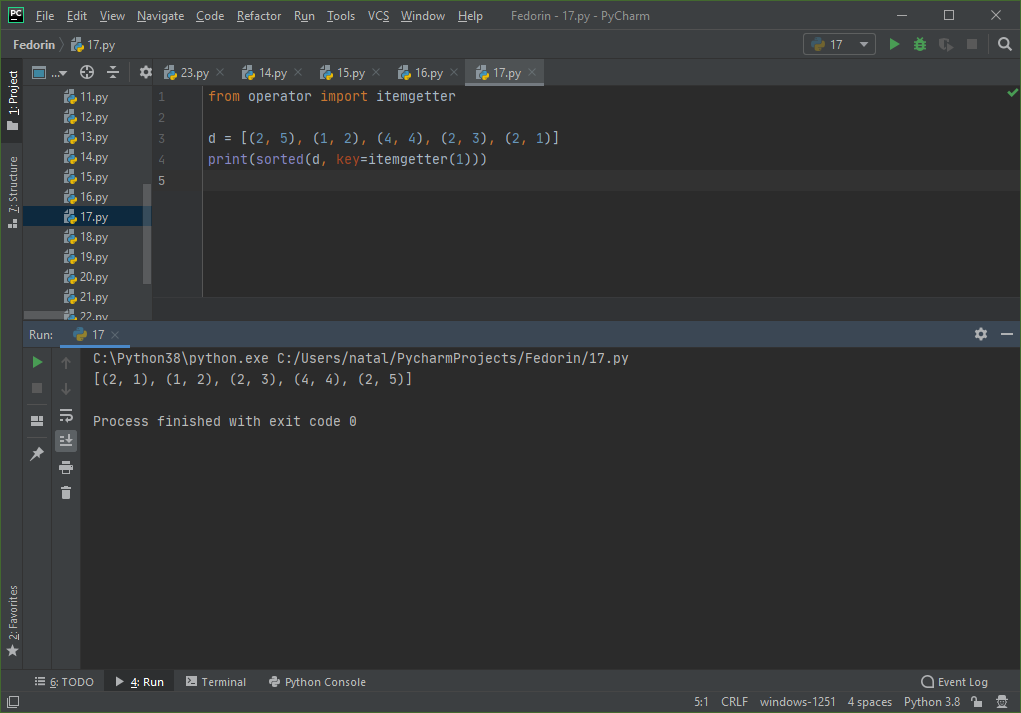


17) Напишіть програму Python, щоб отримати список, відсортований у порядку зростання за останнім елементом у кожному кортежі із заданого списку не порожніх кортежів

**Програмна реалізація:**

from operator import itemgetter  
  
d = [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]  
print(sorted(d, key=itemgetter(1)))

**Результат:**

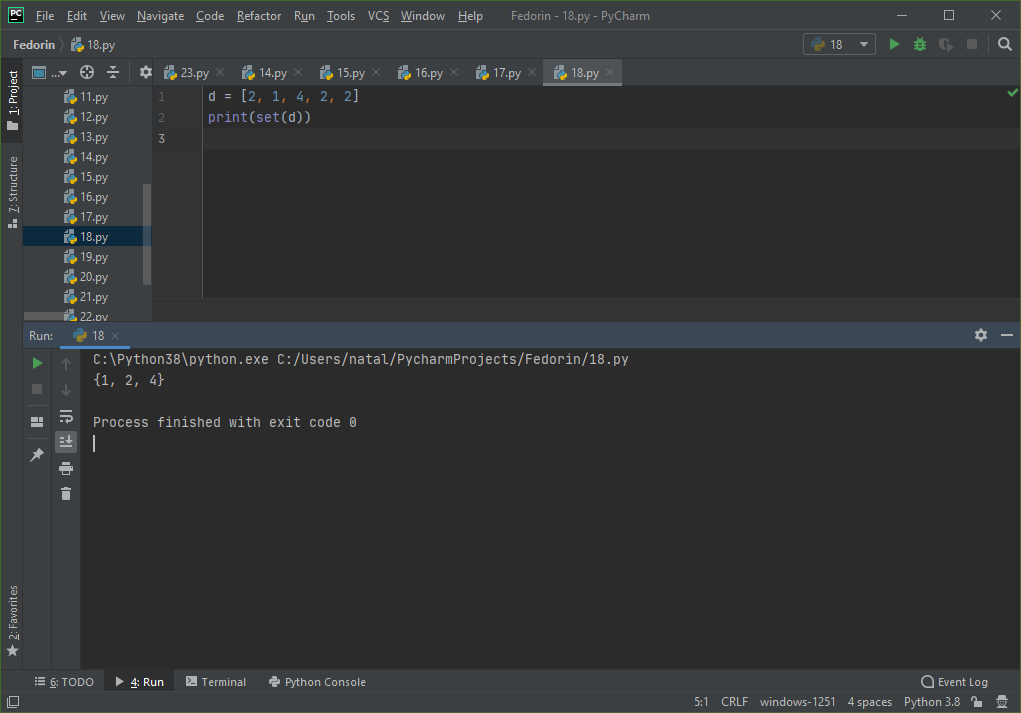


18) Яка видаляє елементі які повторюються

**Програмна реалізація:**

d = [2, 1, 4, 2, 2]  
print(set(d))

**Результат:**

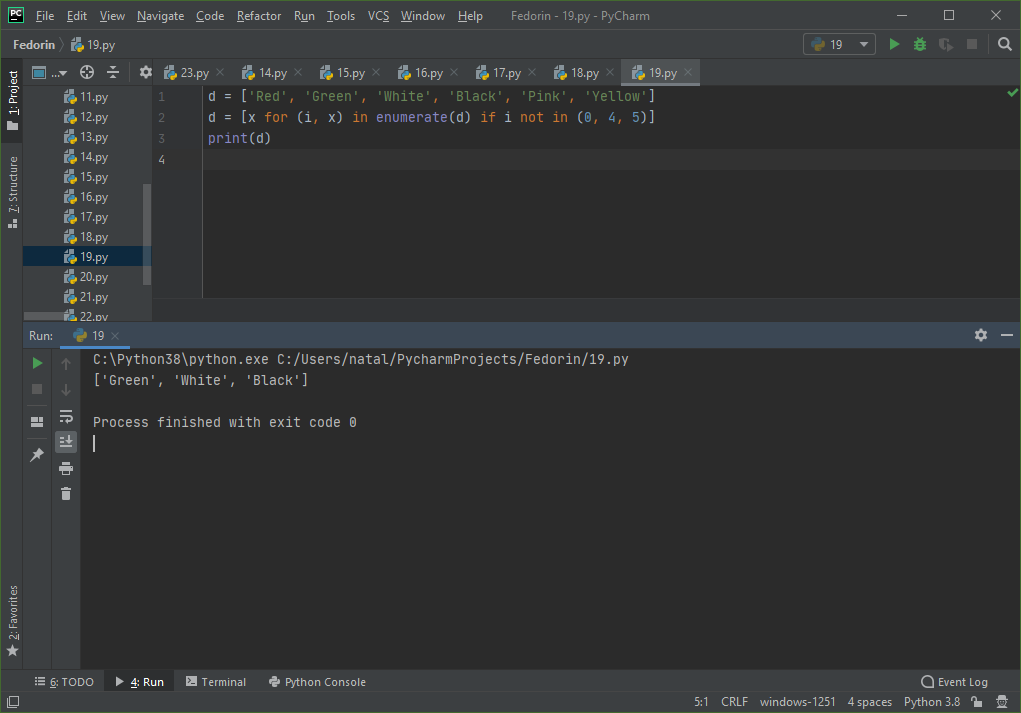


19) Яка друкує лист після видалення 0th, 4th та 5th елементів

**Програмна реалізація:**

d = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']  
d = [x for (i, x) in enumerate(d) if i not in (0, 4, 5)]  
print(d)

**Результат:**

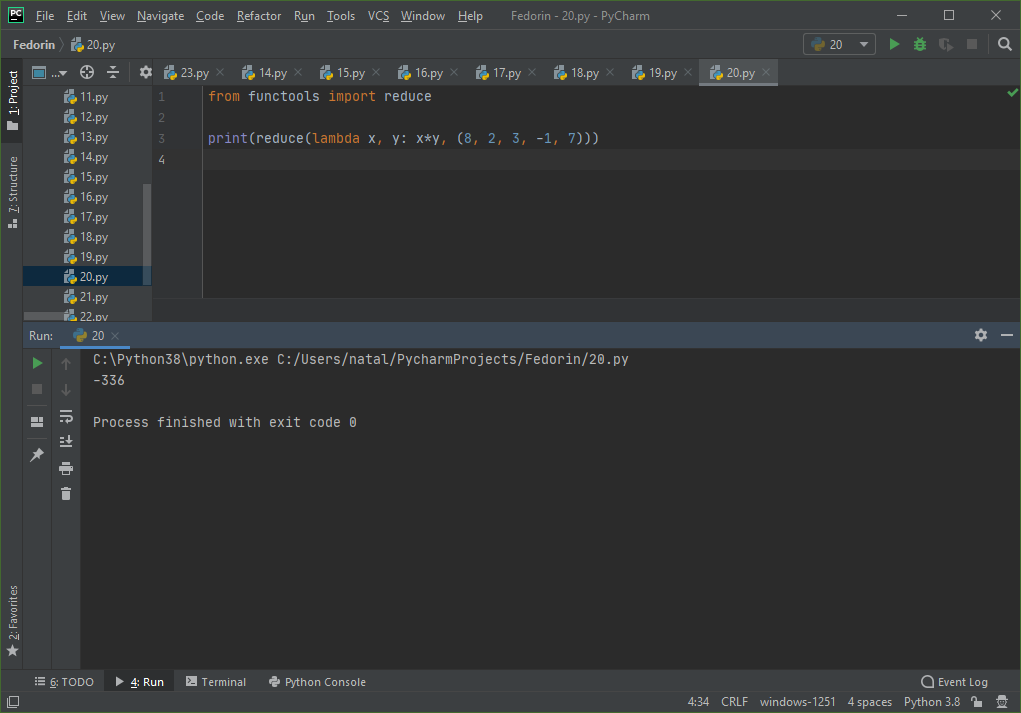


20) Програмка яка помножує всі значення в списку

**Програмна реалізація:**

from functools import reduce  
  
print(reduce(lambda x, y: x\*y, (8, 2, 3, -1, 7)))

**Результат:**

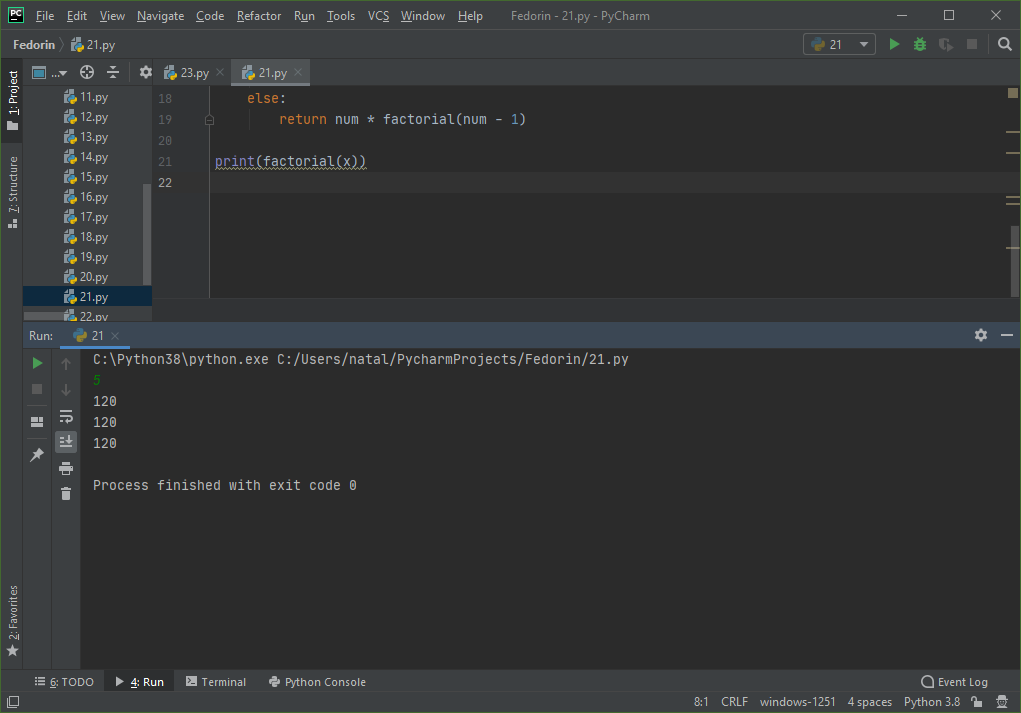


21) Розраховує факторіал числа

**Програмна реалізація:**

import math  
  
x = int(input())  
  
#Math function  
print(math.factorial(x))  
  
#Cycle  
res = 1  
for i in range(1, x + 1):  
 res \*= i  
print(res)  
  
#Recursion  
def factorial(num):  
 if num == 0:  
 return 1  
 else:  
 return num \* factorial(num - 1)  
  
print(factorial(x))

**Результат:**

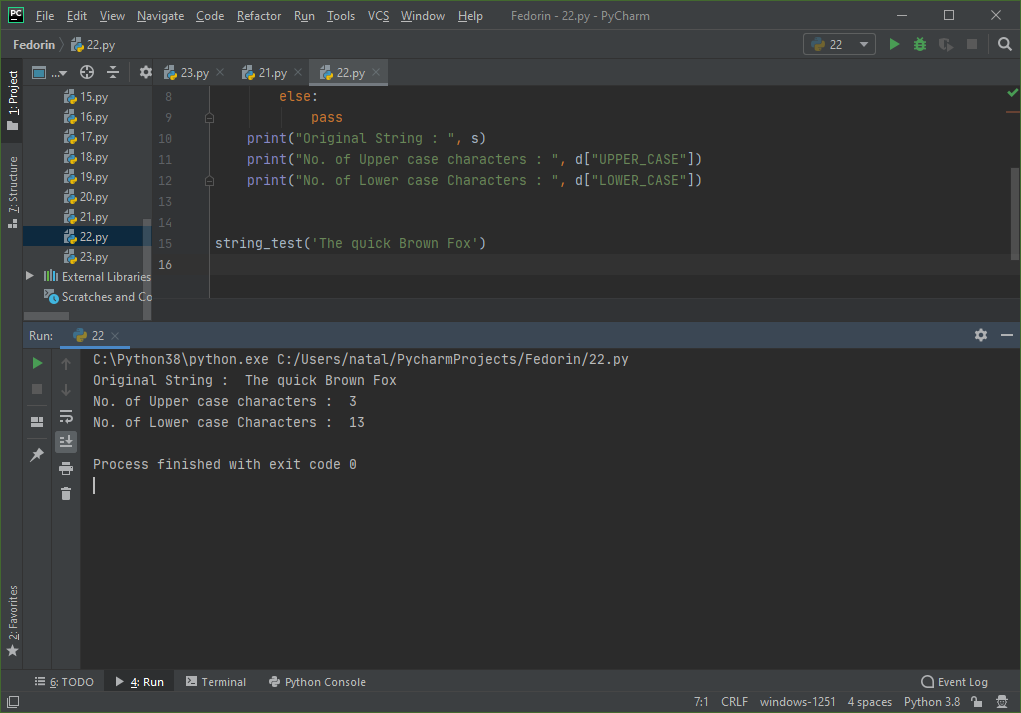


22) Приймає строку та підраховує кількість upper case letters and lower case letters

**Програмна реалізація:**

def string\_test(s):  
 d = {"UPPER\_CASE": 0, "LOWER\_CASE": 0}  
 for c in s:  
 if c.isupper():  
 d["UPPER\_CASE"] += 1  
 elif c.islower():  
 d["LOWER\_CASE"] += 1  
 else:  
 pass  
 print("Original String : ", s)  
 print("No. of Upper case characters : ", d["UPPER\_CASE"])  
 print("No. of Lower case Characters : ", d["LOWER\_CASE"])  
  
  
string\_test('The quick Brown Fox')

**Результат:**



23) Друкує патерн, використовуючи цикли

**Програмна реалізація:**

for i in range(1, 6):  
 print('\* ' \* i)  
  
for i in range(4, 0, -1):  
 print('\* ' \* i)

**Результат:**

