B. K. BIRLA COLLEGE OF ARTS, SCIENCE & COMMERCE (AUTONOMOUS), KALYAN

DEPARTMENT OF INFORMATION TECHNOLOGY

Student Name:	**Adarsh Gupta**
Student ID:	**4824559**
Class:	**MSC IT (CC)**
Subject:	**Data Science Using Python**

B. K. BIRLA COLLEGE OF ARTS, SCIENCE & COMMERCE

(AUTONOMOUS), KALYAN

(Affiliated to University of Mumbai)

KALYAN-MAHARASHTRA-421301

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that Mr ***Adarsh Gupta*** bearing Seat. No: **(4824559)**, in class _has successfully completed practical of the subject ***Data Science Using Python***

Teacher's Signature: _____

Place: Birla College

Date: 4th Jan 2024 **College Seal**

INDEX

SR. NO	PRACTICAL NAME	SIGNATURE
1.	Practical on Basics of Python (Operators)	
2.	Practical on Control and For Loop Statements	
3.	Practical on While loop and Break Statements	
4.	Practical on List, Tuples, Dictionary and Functions	
5.	Operations on Dataset	
6.	Practical on Sets	
7.	Practical on Functions	
8.	Practical on Exception Handling and file handling	
9.	Practical on Regular Expression	

Practical No. 1

Aim: Practical on basics of python

❖ **Write a python program to print employee name, id, salary.** Code & Output:

```
#Prg to print employee name, id, salary.
emp_name="Smita Parale"
emp_id=1
salary=20000
print("Employee name is : ", emp_name)
print("Employee id is : ", emp_id)
print("Employee salary is : ", salary)
```

```
========== RESTART: C:/Users/admin/Desktop/demo.py ==========
Employee name is :  Smita Parale
Employee id is :  1
Employee salary is :  20000
>>>
```

❖ **Write a python program for arithmetic operators and logical operators.**

 # Arithmetic operators Code

& Output:

```
#Prg for arithetic operators and loical operators

#Arithmetic Operators
x=5
y=5
print("Addition is:", x+y)
print("Substraction is:", x-y)
print("Multiplication is:", x*y)
print("Division is:", x/y)
print("Modulus is:", x%y)
print("Exponent is:", x**y)
print("Floor division is:", x//y)
```

```
========== RESTART: C:/Users/admin/Desktop/demo.py ==========
Addition is: 10
Substraction is: 0
Multiplication is: 25
Division is: 1.0
Modulus is: 0
Exponent is: 3125
Floor division is: 1
>>>
```

#Logical operators Code

& Output:

```
#Prg for arithetic operators and loical operators

#Logical Operators
x=5
y=5
print(x==y and x>y)
print(x==y or x>y)
print(not(x>y and x==y))
```

```
False
True
True
>>>
```

❖ **Write a python to solve the following equation: (10*2+40/9-60+2).** Code & Output:

```
#solve the following (10*2+40/9-60+2)

print(10*2+40/9-60+2)
```

```
>>> ================================ RESTART: C:/Users/admin/Desktop/demo.py ================================
-33.55555555555556
>>>
```

❖ **Write a python program to accept value from user like name, salary then calculate the annual income of the employee.** Code & Output:

```
#prg to accept value from user like name and salary then calculate annual income of the employee
emp_name=input("Enter the name:")
salary=int(input("Enter the salary:"))
print("Name of the employee is: "+emp_name)
print("salary of the employee is: ", salary)
print("annual income of the employee is: ",salary*12)
```

```
================================ RESTART: C:/Users/admin/Desktop/demo.py ================================
Enter the name:Smita
Enter the salary:20000
Name of the employee is: Smita
salary of the employee is:  20000
annual income of the employee is:  240000
>>>
```

❖ **Write a python program to accept values from user like amount, time period and rate if interest and then calculate the simple interest.**
Code & Output:

```
#prg to accept value from user like amount, time period and rate of interest then calculate te simple interest
A=int(input("Enter the amount:"))
T=int(input("Enter the Time period:"))
I=int(input("Enter the rate of ineterest:"))
print("Principal Amount is: ",A)
print("Time Perios is: ",T, "yrs")
print("Rate of Interest is: ",I)
print("Simple Interest= ",A*T*I/100)
```

```
================================ RESTART: C:/Users/admin/Desktop/demo.py ================================
Enter the amount:3000
Enter the Time period:2
Enter the rate of ineterest:3
Principal Amount is:  3000
Time Perios is:  2 yrs
Rate of Interest is:  3
Simple Interest=  180.0
>>>
```

Practical No. 2

Aim: Practical on control and for loop statements

❖ **Find the final ticket price with the following conditions:**
a) If men and senior citizen, 70% of fare is applicable

b) If female and senior citizen, 80% of fare is applicable

a) If female and normal citizen, 70% of fare is applicable

a) If men and normal citizen, 100% of fare is applicable Code & Output:

```
gender = input("Enter Gender (M/F):")
senior_citizen = input("Are you a senior citizen? (Y/N):")
fare = float(input("Enter the fare:"))

if gender =="M" and senior_citizen == "Y":
    price= 0.7*fare

elif gender == "F" and senior_citizen == "Y":
    price= 0.8*fare

elif gender == "F" and senior_citizen == "N":
    price= 0.7*fare

elif gender == "M" and senior_citizen == "N":
    price= fare

else:
    print("Invalid input. Please enter the valid input")

print(f"Final ticket price: ${price:.2f}")
```

```
========== RESTART: C:/Users/admin/Desktop/dspy/d.py ==========
Enter Gender (M/F):F
Are you a senior citizen? (Y/N):N
Enter the fare:2000
Final ticket price: $1400.00
>>>
```

❖ **Take a dataset for diabetes, Enter the 50 patient data and use for, if else statement.** Code & Output:

```
dataset =[113,145,155,162,140,141,123,156,126,147,158,169,165,154,198,187,100,195,186,189,200,174,185,196,163,152,149,124,134,158,175,167,120,119,118,155,188
for i in dataset:
    if i>120:
        print("Blood sugar is High")

    elif i<100 and i>120:
        print("Blood sugar is Normal")

    else:
        print("Blood sugra is Low")
```

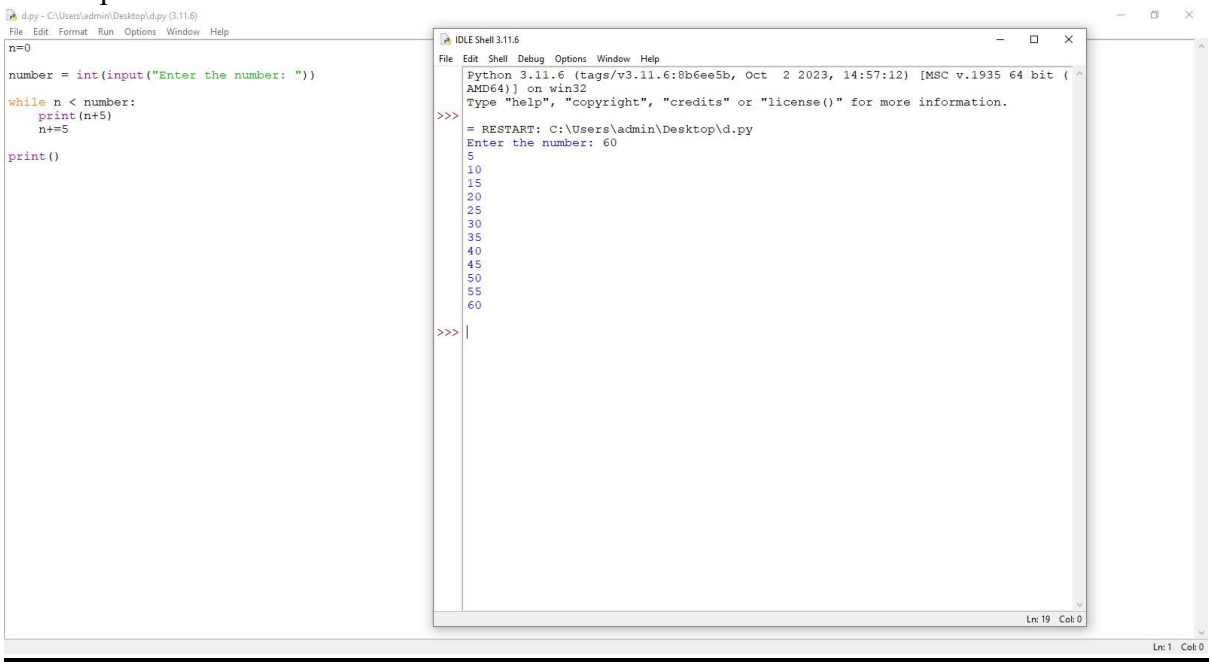
```
Enter the fare:2000
Final ticket price: $1400.00
>>>
========== RESTART: C:/Users/admin/Desktop/dspy/d.py ==========
Blood sugra is Low
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugra is Low
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugra is Low
Blood sugra is Low
Blood sugra is Low
Blood sugar is High
Blood sugar is High
Blood sugar is High
Blood sugar is High
>>>
```

P

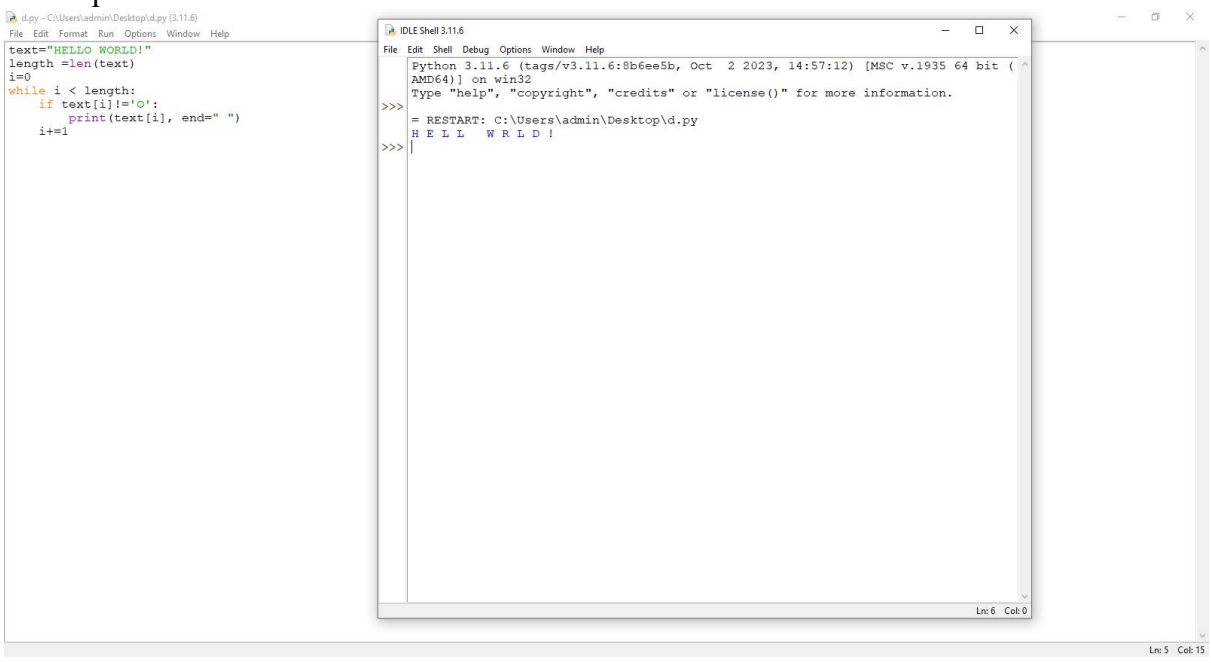
Practical No. 3

Aim: ractical on while loop and break statements

- **Write a python program to find number series by adding +5.** Code & Output:

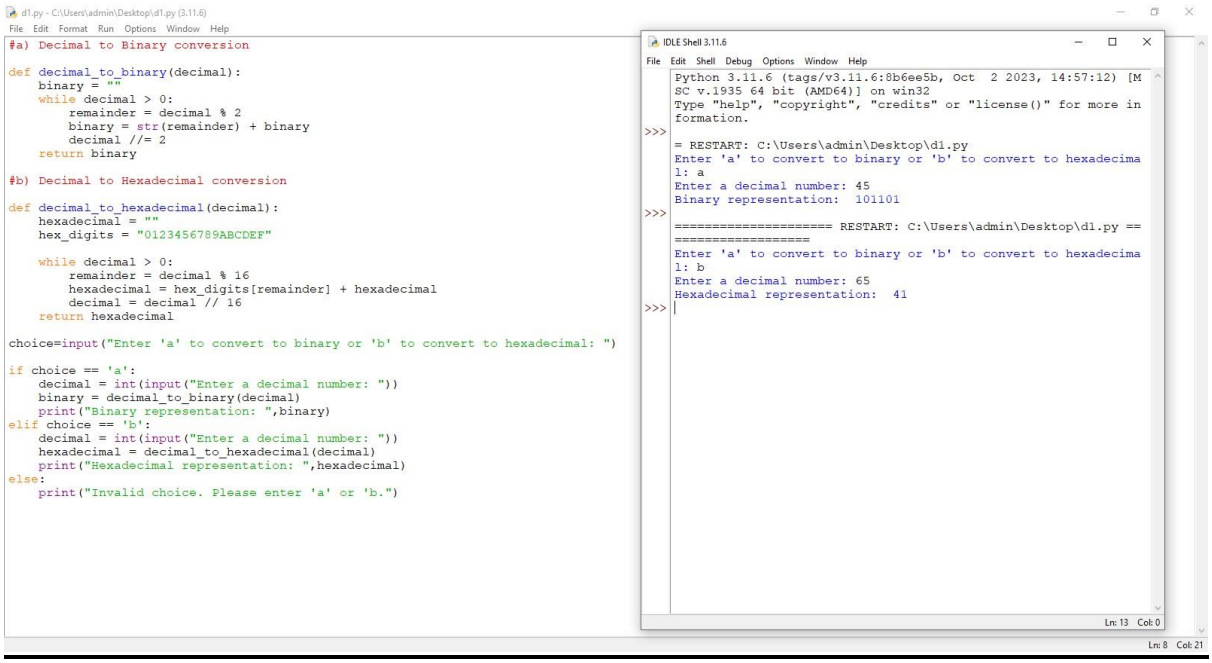


- **Print a string "Hello world" except o using python while loop** Code & Output:



❖ **Convert a number**
a) decimal to binary

b) decimal to hexadecimal
Code & Output:



```
#a) Decimal to Binary conversion

def decimal_to_binary(decimal):
    binary = ""
    while decimal > 0:
        remainder = decimal % 2
        binary = str(remainder) + binary
        decimal //= 2
    return binary

#b) Decimal to Hexadecimal conversion

def decimal_to_hexadecimal(decimal):
    hexadecimal = ""
    hex_digits = "0123456789ABCDEF"

    while decimal > 0:
        remainder = decimal % 16
        hexadecimal = hex_digits[remainder] + hexadecimal
        decimal = decimal // 16
    return hexadecimal

choice=input("Enter 'a' to convert to binary or 'b' to convert to hexadecimal: ")

if choice == 'a':
    decimal = int(input("Enter a decimal number: "))
    binary = decimal_to_binary(decimal)
    print("Binary representation: ",binary)
elif choice == 'b':
    decimal = int(input("Enter a decimal number: "))
    hexadecimal = decimal_to_hexadecimal(decimal)
    print("Hexadecimal representation: ",hexadecimal)
else:
    print("Invalid choice. Please enter 'a' or 'b.")
```

```
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [M
SC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in
formation.
>>>
= RESTART: C:\Users\admin\Desktop\d1.py
Enter 'a' to convert to binary or 'b' to convert to hexadecima
l: a
Enter a decimal number: 45
Binary representation:  101101
>>>
========== RESTART: C:\Users\admin\Desktop\d1.py ==
==========
Enter 'a' to convert to binary or 'b' to convert to hexadecima
l: b
Enter a decimal number: 65
Hexadecimal representation:  41
>>>
```

❖ **Find the average of given number from user after finding average reverse the number.** Code & Output:

P

```
def reverse_number(num):
    reversed_num = 0
    while num > 0:
        digit = num % 10
        reversed_num = reversed_num * 10 + digit
        num //= 10
    return reversed_num

numbers = []

n = int(input("Enter the number of values: "))

for i in range(n):
    num = float(input(f"Enter value {i + 1}: "))
    numbers.append(num)


average = sum(numbers) / n
print("The average of the" ,n, "numbers is:", average)

reversed_average = reverse_number(int(average))
print(f"Reversed average: {reversed_average}")
```

```
========== RESTART: C:\Users\admin\Desktop\d.py ==========
Enter the number of values: 4
Enter value 1: 10
Enter value 2: 20
Enter value 3: 65
Enter value 4: 45
The average of the 4 numbers is: 35.0
Reversed average: 53
>>>
```

Practical No. 4 Aim:

ractical on list, tuples, dictionary and functions

- **Function in Python:**
- **Write a program to calculate the addition of two numbers.**
  Code & Output:

```
def addition(a,b):
    c=a+b
    return c

print("Addition = ",addition(30,40))
```

```
========== RESTART: C:\Users\admin\Desktop\list.py ==========
Addition =  70
>>>
```

- **Write a program to find the annual salary of an employee.** Code & Output:

```
def annual_salary(salary):
    annual_sal=salary*12
    return annual_sal

print("Annual Salary =", annual_salary(50000))
```

```
========== RESTART: C:\Users\admin\Desktop\list.py ==========
Annual Salary = 600000
>>>
```

- **Write a program to find the rate of interest for car loan, home loan and education.**
  Code:

list.py - C:\Users\admin\Desktop\list.py (3.11.6)

File Edit Format Run Options Window Help

```
def loan(total_interest, principal, yrs):
    rate = total_interest*100 / principal*yrs
    return rate

print("Car Loan \nHome Loan \nEducation Loan")
choice=int(input("Enter your choice:"))

if choice==1:
    total_interest=int(input("Enter the total interest paid :"))
    principal=int(input("Enter the principal amount:"))
    yrs=int(input("Enter the number of years:"))
    print("Rate of interest of your car is:", loan(total_interest, principal, yrs),"%")

if choice==2:
    total_interest=int(input("Enter the total interest paid :"))
    principal=int(input("Enter the principal amount:"))
    yrs=int(input("Enter the number of years:"))
    print("Rate of interest of your home is:", loan(total_interest, principal, yrs),"%")

if choice==3:
    total_interest=int(input("Enter the total interest paid :"))
    principal=int(input("Enter the principal amount:"))
    yrs=int(input("Enter the number of years:"))
    print("Rate of interest of your education is:", loan(total_interest, principal, yrs),"%")

else:
    print("You have made wrong choice")
```

Output:

IDLE Shell 3.11.6

File Edit Shell Debug Options Window Help

```
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\admin\Desktop\list.py
Car Loan
Home Loan
Education Loan
Enter your choice:3
Enter the total interest paid :5000
Enter the principal amount:200000
Enter the number of years:2
Rate of interest of your education is: 5.0 %
>>>
```

P

Practical No. 6

Aim: ractical on sets

❖ Sets operation on Agriculture Dataset in Python:

a. Creating Set:

Code:

```
Agriculture_dataset_sets.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py (3.11.6)
File  Edit  Format  Run  Options  Window  Help
{('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
{('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
{('Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)),
{('Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
{('Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)),
{('Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)),
{('Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)),
{('Farm_ID', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)),
{('Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)),
{('Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)),
{('Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)),
{('Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)),
{('Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)),
{('Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)),
{('Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)),
{('Farm_ID', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)),
{('Farm_ID:', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)),
{('Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)),
{('Farm_ID:', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)),
{('Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)),
{('Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)),
{('Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)),
{('Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
{('Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)),
{('Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)),
{('Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))}

agricultural_dataset2 ={
   {('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
   {('Farm_ID', 31), ('Crop', 'Spinach'), ('Acres_Planted', 15), ('Yield_Per_Acre', 90), ('Total_Production', 1350), ('Market_Price', 18), ('Revenue', 24300), ('Expense', 10000), ('Profit', 14300)),
   {('Farm_ID', 33), ('Crop', 'Kale'), ('Acres_Planted', 35), ('Yield_Per_Acre', 80), ('Total_Production', 2800), ('Market_Price', 15), ('Revenue', 42000), ('Expense', 15000), ('Profit', 27000)),
   {('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
   }

print(agricultural_data.intersection(agricultural_dataset2))
```

Output:

```
= RESTART: D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py
{{('Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000)), {('Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90),
('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)), {('Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Marke
t_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)), {('Farm_ID:', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profi
t', 50600)), {('Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)), {('Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Pla
nted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)), {('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Producti
on', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)), {('Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Exp
ense', 35000), ('Profit', 105000)), {('Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)), {('Farm_ID', 22), ('Crop',
'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)), {('Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('
Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)), {('Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue
', 44100), ('Expense', 17000), ('Profit', 27100)), {('Farm_ID:', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)), {('Farm_ID:', 2),
('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)), {('Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35),
('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)), {('Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenu
e', 41250), ('Expense', 12000), ('Profit', 29250)), {('Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)), {('F
arm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)), {('Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yi
eld_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)), {('Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Pri
ce', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000)), {('Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 1045
0)), {('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)), {('Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 7
0), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)), {('Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Mark
et_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000)), {('Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('
Profit', 42000)), {('Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)), {('Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_
Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)), {('Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Productio
n', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)), {('Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('E
xpense', 40000), ('Profit', 80000)), {('Farm_ID:', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000))}
>>>
```

b. Access the Dataset items:

Code:

```
Agriculture_dataset_sets.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py (3.11.6)
File  Edit  Format  Run  Options  Window  Help
agricultural_data = {
{('Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000)),
{('Farm_ID:', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000)),
{('Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)),
{('Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000)),
{('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
{('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
{('Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)),
{('Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
{('Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)),
{('Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)),
{('Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)),
{('Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)),
{('Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)),
{('Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)),
{('Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)),
{('Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)),
{('Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)),
{('Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)),
{('Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)),
{('Farm_ID', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)),
{('Farm_ID', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)),
{('Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)),
{('Farm_ID', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)),
{('Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)),
{('Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)),
{('Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)),
{('Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
{('Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)),
{('Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)),
{('Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))}


for data in agricultural_data:
   print(data)
```

Output:

```
IDLE Shell 3.11.6
File  Edit  Shell  Debug  Options  Window  Help
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py
{('Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000)}
{('Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)}
{('Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)}
{('Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)}
{('Farm_ID:', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)}
{('Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)}
{('Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)}
{('Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)}
{('Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)}
{('Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)}
{('Farm_ID', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)}
{('Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000)}
{('Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)}
{('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)}
{('Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)}
{('Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)}
{('Farm_ID:', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000)}
{('Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)}
{('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)}
{('Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)}
{('Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000)}
{('Farm_ID:', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)}
{('Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)}
{('Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)}
{('Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)}
{('Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)}
{('Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)}
{('Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)}
{('Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)}
{('Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)}
>>>
```

c. Add items in Dataset using add():

Code:

```
Agriculture_dataset_sets.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py (3.11.6)
File  Edit  Format  Run  Options  Window  Help
agricultural_data = {
{('Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000)),
{('Farm_ID:', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000)),
{('Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)),
{('Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000)),
{('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
{('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
{('Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)),
{('Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
{('Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)),
{('Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)),
{('Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)),
{('Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)),
{('Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)),
{('Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)),
{('Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)),
{('Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)),
{('Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)),
{('Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)),
{('Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)),
{('Farm_ID', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)),
{('Farm_ID', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)),
{('Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)),
{('Farm_ID', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)),
{('Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)),
{('Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)),
{('Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)),
{('Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
{('Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)),
{('Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)),
{('Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000)}}

agricultural_data.add({('Farm_ID', 31), ('Crop', 'Peas'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 10), ('Revenue', 24000), ('Expense', 9000), ('Profit', 15000)})

for data in agricultural_data:
    print(data)

print(len(agricultural_data))
```

Output:

```
IDLE Shell 3.11.6
File  Edit  Shell  Debug  Options  Window  Help
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py
{('Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)}
{('Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)}
{('Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)}
{('Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000)}
{('Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)}
{('Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)}
{('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)}
{('Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)}
{('Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)}
{('Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)}
{('Farm_ID:', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000)}
{('Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)}
{('Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)}
{('Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)}
{('Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)}
{('Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000)}
{('Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)}
{('Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)}
{('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)}
{('Farm_ID:', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)}
{('Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)}
{('Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)}
{('Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)}
{('Farm_ID:', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)}
{('Farm_ID', 31), ('Crop', 'Peas'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 10), ('Revenue', 24000), ('Expense', 9000), ('Profit', 15000)}
{('Farm_ID:', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)}
{('Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)}
{('Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)}
{('Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)}
{('Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)}
{('Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000)}
31
```

d. Add sets using update():

Code:

```
Agriculture_dataset_sets.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py (3.11.6)
File  Edit  Format  Run  Options  Window  Help
({'Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)),
({'Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
({'Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)),
({'Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)),
({'Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)),
({'Farm_ID', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)),
({'Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)),
({'Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)),
({'Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)),
({'Farm_ID', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)),
({'Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)),
({'Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)),
({'Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)),
({'Farm_ID', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)),
({'Farm_ID', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)),
({'Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)),
({'Farm_ID', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)),
({'Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)),
({'Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)),
({'Farm_ID', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)),
({'Farm_ID', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
({'Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)),
({'Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)),
({'Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))}

agricultural_dataset2={
   ({'Farm_ID', 35), ('Crop', 'Artichokes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 70), ('Total_Production', 1750), ('Market_Price', 20), ('Revenue', 35000), ('Expense', 12000), ('Profit', 23000)),
   ({'Farm_ID', 40), ('Crop', 'Raspberries'), ('Acres_Planted', 55), ('Yield_Per_Acre', 100), ('Total_Production', 5500), ('Market_Price', 22), ('Revenue', 121000), ('Expense', 25000), ('Profit', 96000)),
   ({'Farm_ID', 39), ('Crop', 'Kiwifruit'), ('Acres_Planted', 45), ('Yield_Per_Acre', 75), ('Total_Production', 3375), ('Market_Price', 15), ('Revenue', 50625), ('Expense', 20000), ('Profit', 30625))
   }

agricultural_data.update(agricultural_dataset2)

for data in agricultural_data:
   print(data)

print(len(agricultural_data))
```

Output:

```
>>>
   = RESTART: D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py
   ({'Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100))
   ({'Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200))
   ({'Farm_ID', 39), ('Crop', 'Kiwifruit'), ('Acres_Planted', 45), ('Yield_Per_Acre', 75), ('Total_Production', 3375), ('Market_Price', 15), ('Revenue', 50625), ('Expense', 20000), ('Profit', 30625))
   ({'Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950))
   ({'Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000))
   ({'Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000))
   ({'Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500))
   ({'Farm_ID', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000))
   ({'Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000))
   ({'Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000))
   ({'Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600))
   ({'Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250))
   ({'Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000))
   ({'Farm_ID', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400))
   ({'Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125))
   ({'Farm_ID', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300))
   ({'Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800))
   ({'Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000))
   ({'Farm_ID', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000))
   ({'Farm_ID', 40), ('Crop', 'Raspberries'), ('Acres_Planted', 55), ('Yield_Per_Acre', 100), ('Total_Production', 5500), ('Market_Price', 22), ('Revenue', 121000), ('Expense', 25000), ('Profit', 96000))
   ({'Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000))
   ({'Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000))
   ({'Farm_ID', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000))
   ({'Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))
   ({'Farm_ID', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600))
   ({'Farm_ID', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000))
   ({'Farm_ID', 35), ('Crop', 'Artichokes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 70), ('Total_Production', 1750), ('Market_Price', 20), ('Revenue', 35000), ('Expense', 12000), ('Profit', 23000))
   ({'Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250))
   ({'Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000))
   ({'Farm_ID', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600))
   ({'Farm_ID', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000))
   ({'Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450))
   ({'Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500))
   33
```

e. Remove items:

remove():

Code:

```
*Agriculture_dataset_sets.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py (3.11.6)*
File  Edit  Format  Run  Options  Window  Help
agricultural_data = {
({'Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000)),
({'Farm_ID', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000)),
({'Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)),
({'Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000)),
({'Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
({'Farm_ID', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
({'Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)),
({'Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
({'Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)),
({'Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)),
({'Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)),
({'Farm_ID', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)),
({'Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)),
({'Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)),
({'Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)),
({'Farm_ID', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)),
({'Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)),
({'Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)),
({'Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)),
({'Farm_ID', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)),
({'Farm_ID', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)),
({'Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)),
({'Farm_ID', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)),
({'Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)),
({'Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)),
({'Farm_ID', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)),
({'Farm_ID', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
({'Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)),
({'Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)),
({'Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))}

agricultural_data.remove(({'Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)))

for data in agricultural_data:
   print(data)

print(len(agricultural_data))
```

Output:

discard():

Code:

Output:

f. pop():

Code:

```
Agriculture_dataset_sets.py - D:\SMITA\MSc IT CC P1\DSPY\Programs\Agriculture_dataset_sets.py (3.11.6)
File  Edit  Format  Run  Options  Window  Help
agricultural_data = {
({'Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000)),
({'Farm_ID:', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000)),
({'Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)),
({'Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000)),
({'Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
({'Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
({'Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)),
({'Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
({'Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)),
({'Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)),
({'Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)),
({'Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)),
({'Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)),
({'Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)),
({'Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)),
({'Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)),
({'Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)),
({'Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)),
({'Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)),
({'Farm_ID:', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)),
({'Farm_ID:', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)),
({'Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)),
({'Farm_ID:', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)),
({'Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)),
({'Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)),
({'Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)),
({'Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
({'Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)),
({'Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)),
({'Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))]

agricultural_data.pop()
print(len(agricultural_data))
```

Output:

```
>>>
= RESTART: D:\SMITA\MSc IT CC P1\DSPY\Programs\Agriculture_dataset_sets.py
29
```

g. clear():

Code:

```
Agriculture_dataset_sets.py - D:\SMITA\MSc IT CC P1\DSPY\Programs\Agriculture_dataset_sets.py (3.11.6)
File  Edit  Format  Run  Options  Window  Help
agricultural_data = {
({'Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000)),
({'Farm_ID:', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000)),
({'Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)),
({'Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000)),
({'Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
({'Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
({'Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)),
({'Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
({'Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)),
({'Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)),
({'Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)),
({'Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)),
({'Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)),
({'Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)),
({'Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)),
({'Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)),
({'Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)),
({'Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)),
({'Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)),
({'Farm_ID:', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)),
({'Farm_ID:', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)),
({'Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)),
({'Farm_ID:', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)),
({'Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)),
({'Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)),
({'Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)),
({'Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
({'Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)),
({'Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)),
({'Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))]

agricultural_data.clear()
print(len(agricultural_data))
```

Output:

```
IDLE Shell 3.11.6
File  Edit  Shell  Debug  Options  Window  Help
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win3
2
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\SMITA\MSc IT CC P1\DSPY\Programs\Agriculture_dataset_sets.py
0
```

h. union():

Code:

```
Agriculture_dataset_sets.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py (3.11.6)
File  Edit  Format  Run  Options  Window  Help
agricultural_data = {
(('Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000)),
(('Farm_ID:', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000)),
(('Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)),
(('Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000)),
(('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)
(('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
(('Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)),
(('Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
(('Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)),
(('Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)),
(('Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)),
(('Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)),
(('Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)),
(('Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)
(('Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)),
(('Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)),
(('Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 800
(('Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)),
(('Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)),
(('Farm_ID:', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)),
(('Farm_ID:', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)),
(('Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)),
(('Farm_ID:', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)),
(('Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 650
(('Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)),
(('Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)),
(('Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
(('Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)),
(('Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)),
(('Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))}

agricultural_dataset2={
    (('Farm_ID', 36), ('Crop', 'Asparagus'), ('Acres_Planted', 30), ('Yield_Per_Acre', 50), ('Total_Production', 1500), ('Market_Price', 25), ('Revenue', 37500), ('Expense', 15000), ('Profit', 225
(('Farm_ID', 31), ('Crop', 'Spinach'), ('Acres_Planted', 15), ('Yield_Per_Acre', 90), ('Total_Production', 1350), ('Market_Price', 18), ('Revenue', 24300), ('Expense', 10000), ('Profit', 14300)),
(('Farm_ID', 33), ('Crop', 'Kale'), ('Acres_Planted', 35), ('Yield_Per_Acre', 80), ('Total_Production', 2800), ('Market_Price', 15), ('Revenue', 42000), ('Expense', 15000), ('Profit', 27000)),
(('Farm_ID', 38), ('Crop', 'Plums'), ('Acres_Planted', 20), ('Yield_Per_Acre', 120), ('Total_Production', 2400), ('Market_Price', 25), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
(('Farm_ID', 35), ('Crop', 'Artichokes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 70), ('Total_Production', 1750), ('Market_Price', 20), ('Revenue', 35000), ('Expense', 12000), ('Profit', 23000)
(('Farm_ID', 40), ('Crop', 'Raspberries'), ('Acres_Planted', 55), ('Yield_Per_Acre', 100), ('Total_Production', 5500), ('Market_Price', 22), ('Revenue', 121000), ('Expense', 25000), ('Profit', 960
(('Farm_ID', 39), ('Crop', 'Kiwifruit'), ('Acres_Planted', 45), ('Yield_Per_Acre', 75), ('Total_Production', 3375), ('Market_Price', 15), ('Revenue', 50625), ('Expense', 20000), ('Profit', 30625)),
(('Farm_ID', 37), ('Crop', 'Mangoes'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 30), ('Revenue', 96000), ('Expense', 30000), ('Profit', 66000)),
(('Farm_ID', 32), ('Crop', 'Chickpeas'), ('Acres_Planted', 50), ('Yield_Per_Acre', 25), ('Total_Production', 1250), ('Market_Price', 12), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
(('Farm_ID', 34), ('Crop', 'Peas'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 10), ('Revenue', 24000), ('Expense', 9000), ('Profit', 15000))
    }

agricultural_dataset3=agricultural_data.union(agricultural_dataset2)

for data in agricultural_dataset3:
    print(data)
```

Ln: 49 Col: 14

Output:

```
IDLE Shell 3.11.6
File  Edit  Shell  Debug  Options  Window  Help
    Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    = RESTART: D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py
    (('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000
    ))
    (('Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250
    ))
    (('Farm_ID', 39), ('Crop', 'Kiwifruit'), ('Acres_Planted', 45), ('Yield_Per_Acre', 75), ('Total_Production', 3375), ('Market_Price', 15), ('Revenue', 50625), ('Expense', 20000), ('Profit', 306
    25))
    (('Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27
    250))
    (('Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)
    )
    (('Farm_ID', 32), ('Crop', 'Chickpeas'), ('Acres_Planted', 50), ('Yield_Per_Acre', 25), ('Total_Production', 1250), ('Market_Price', 12), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000
    ))
    (('Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500
    ))
    (('Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit',
    65000))
    (('Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000))
    (('Farm_ID', 35), ('Crop', 'Artichokes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 70), ('Total_Production', 1750), ('Market_Price', 20), ('Revenue', 35000), ('Expense', 12000), ('Profit', 23
    000))
    (('Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 1300
    0))
    (('Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000))
    (('Farm_ID', 33), ('Crop', 'Kale'), ('Acres_Planted', 35), ('Yield_Per_Acre', 80), ('Total_Production', 2800), ('Market_Price', 15), ('Revenue', 42000), ('Expense', 15000), ('Profit', 27000))
    (('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 2
    2000))
    (('Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 272
    00))
    (('Farm_ID', 36), ('Crop', 'Asparagus'), ('Acres_Planted', 30), ('Yield_Per_Acre', 50), ('Total_Production', 1500), ('Market_Price', 25), ('Revenue', 37500), ('Expense', 15000), ('Profit', 225
    00))
    (('Farm_ID', 40), ('Crop', 'Raspberries'), ('Acres_Planted', 55), ('Yield_Per_Acre', 100), ('Total_Production', 5500), ('Market_Price', 22), ('Revenue', 121000), ('Expense', 25000), ('Profit',
    96000))
    (('Farm_ID:', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400))
    (('Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 420
    00))
    (('Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100
    ))
    (('Farm_ID', 37), ('Crop', 'Mangoes'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 30), ('Revenue', 96000), ('Expense', 30000), ('Profit', 66000
    ))
    (('Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950
    ))
    (('Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit',
    80000))
    (('Farm_ID', 34), ('Crop', 'Peas'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 10), ('Revenue', 24000), ('Expense', 9000), ('Profit', 15000))
    (('Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)
    )
    (('Farm_ID:', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000))
    (('Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600))
    (('Farm_ID:', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000
    ))
    (('Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 1860
    0))
    (('Farm_ID:', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)
    )
    (('Farm_ID', 31), ('Crop', 'Spinach'), ('Acres_Planted', 15), ('Yield_Per_Acre', 90), ('Total_Production', 1350), ('Market_Price', 18), ('Revenue', 24300), ('Expense', 10000), ('Profit', 14300
    ))
    (('Farm_ID', 38), ('Crop', 'Plums'), ('Acres_Planted', 20), ('Yield_Per_Acre', 120), ('Total_Production', 2400), ('Market_Price', 25), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)
    )
    (('Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000
    ))
    (('Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450))
    (('Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 800
    00))
    (('Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500
    ))
    (('Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 7500
    0))
    (('Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))
    (('Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)
    )
    (('Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125))
>>>
```

Ln: 45 Col: 0

i. intersection():

Code:

Agriculture_dataset_sets.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py (3.11.6)

File Edit Format Run Options Window Help

```
(('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
(('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
(('Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)),
(('Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
(('Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)),
(('Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)),
(('Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)),
(('Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)),
(('Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)),
(('Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)),
(('Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)),
(('Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)),
(('Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)),
(('Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)),
(('Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)),
(('Farm_ID:', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)),
(('Farm_ID:', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)),
(('Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)),
(('Farm_ID:', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)),
(('Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)),
(('Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)),
(('Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)),
(('Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
(('Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)),
(('Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)),
(('Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))}

agricultural_dataset2 ={
   (('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
   (('Farm_ID', 31), ('Crop', 'Spinach'), ('Acres_Planted', 15), ('Yield_Per_Acre', 90), ('Total_Production', 1350), ('Market_Price', 18), ('Revenue', 24300), ('Expense', 10000), ('Profit', 14300)),
   (('Farm_ID', 33), ('Crop', 'Kale'), ('Acres_Planted', 35), ('Yield_Per_Acre', 80), ('Total_Production', 2800), ('Market_Price', 15), ('Revenue', 42000), ('Expense', 15000), ('Profit', 27000)),
   (('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
   }

print(agricultural_data.intersection(agricultural_dataset2))
```

Output:

```
>>>
= RESTART: D:/SMITA/MSc IT CC P1/DSPY/Programs/Agriculture_dataset_sets.py
{(('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)), (('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000))}
>>>
```

j. symmetric_difference_update():

Code:

Agriculture_dataset_sets.py - D:\SMITA\MSc IT CC P1\DSPY\Programs\Agriculture_dataset_sets.py (3.11.6)

File Edit Format Run Options Window Help

```
agricultural_data = {
(('Farm_ID', 23), ('Crop', 'Onions'), ('Acres_Planted', 55), ('Yield_Per_Acre', 50), ('Total_Production', 2750), ('Market_Price', 8), ('Revenue', 22000), ('Expense', 11000), ('Profit', 11000)),
(('Farm_ID:', 8), ('Crop', 'Oranges'), ('Acres_Planted', 40), ('Yield_Per_Acre', 80), ('Total_Production', 3200), ('Market_Price', 15), ('Revenue', 48000), ('Expense', 20000), ('Profit', 28000)),
(('Farm_ID', 25), ('Crop', 'Cabbage'), ('Acres_Planted', 25), ('Yield_Per_Acre', 80), ('Total_Production', 2000), ('Market_Price', 10), ('Revenue', 20000), ('Expense', 9000), ('Profit', 11000)),
(('Farm_ID', 26), ('Crop', 'Coffee'), ('Acres_Planted', 15), ('Yield_Per_Acre', 200), ('Total_Production', 3000), ('Market_Price', 35), ('Revenue', 105000), ('Expense', 30000), ('Profit', 75000)),
(('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
(('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),
(('Farm_ID', 15), ('Crop', 'Cucumbers'), ('Acres_Planted', 45), ('Yield_Per_Acre', 80), ('Total_Production', 3600), ('Market_Price', 12), ('Revenue', 43200), ('Expense', 16000), ('Profit', 27200)),
(('Farm_ID', 11), ('Crop', 'Tomatoes'), ('Acres_Planted', 25), ('Yield_Per_Acre', 120), ('Total_Production', 3000), ('Market_Price', 20), ('Revenue', 60000), ('Expense', 18000), ('Profit', 42000)),
(('Farm_ID', 10), ('Crop', 'Carrots'), ('Acres_Planted', 70), ('Yield_Per_Acre', 60), ('Total_Production', 4200), ('Market_Price', 9), ('Revenue', 37800), ('Expense', 15000), ('Profit', 22800)),
(('Farm_ID', 16), ('Crop', 'Oats'), ('Acres_Planted', 95), ('Yield_Per_Acre', 30), ('Total_Production', 2850), ('Market_Price', 7), ('Revenue', 19950), ('Expense', 9500), ('Profit', 10450)),
(('Farm_ID', 28), ('Crop', 'Hops'), ('Acres_Planted', 35), ('Yield_Per_Acre', 100), ('Total_Production', 3500), ('Market_Price', 40), ('Revenue', 140000), ('Expense', 35000), ('Profit', 105000)),
(('Farm_ID:', 6), ('Crop', 'Potatoes'), ('Acres_Planted', 50), ('Yield_Per_Acre', 50), ('Total_Production', 2500), ('Market_Price', 10), ('Revenue', 25000), ('Expense', 12000), ('Profit', 13000)),
(('Farm_ID', 22), ('Crop', 'Broccoli'), ('Acres_Planted', 40), ('Yield_Per_Acre', 60), ('Total_Production', 2400), ('Market_Price', 14), ('Revenue', 33600), ('Expense', 15000), ('Profit', 18600)),
(('Farm_ID', 14), ('Crop', 'Sunflowers'), ('Acres_Planted', 75), ('Yield_Per_Acre', 25), ('Total_Production', 1875), ('Market_Price', 22), ('Revenue', 41250), ('Expense', 14000), ('Profit', 27250)),
(('Farm_ID', 13), ('Crop', 'Peppers'), ('Acres_Planted', 35), ('Yield_Per_Acre', 90), ('Total_Production', 3150), ('Market_Price', 14), ('Revenue', 44100), ('Expense', 17000), ('Profit', 27100)),
(('Farm_ID:', 3), ('Crop', 'Rice'), ('Acres_Planted', 80), ('Yield_Per_Acre', 40), ('Total_Production', 3200), ('Market_Price', 8), ('Revenue', 25600), ('Expense', 10000), ('Profit', 15600)),
(('Farm_ID', 29), ('Crop', 'Blueberries'), ('Acres_Planted', 50), ('Yield_Per_Acre', 120), ('Total_Production', 6000), ('Market_Price', 18), ('Revenue', 108000), ('Expense', 28000), ('Profit', 80000)),
(('Farm_ID', 30), ('Crop', 'Avocado'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 40), ('Revenue', 120000), ('Expense', 40000), ('Profit', 80000)),
(('Farm_ID', 19), ('Crop', 'Peanuts'), ('Acres_Planted', 65), ('Yield_Per_Acre', 35), ('Total_Production', 2275), ('Market_Price', 18), ('Revenue', 40950), ('Expense', 12000), ('Profit', 28950)),
(('Farm_ID:', 9), ('Crop', 'Grapes'), ('Acres_Planted', 60), ('Yield_Per_Acre', 70), ('Total_Production', 4200), ('Market_Price', 18), ('Revenue', 75600), ('Expense', 25000), ('Profit', 50600)),
(('Farm_ID:', 2), ('Crop', 'Corn'), ('Acres_Planted', 150), ('Yield_Per_Acre', 25), ('Total_Production', 3750), ('Market_Price', 4), ('Revenue', 15000), ('Expense', 9000), ('Profit', 6000)),
(('Farm_ID', 27), ('Crop', 'Tobacco'), ('Acres_Planted', 90), ('Yield_Per_Acre', 30), ('Total_Production', 2700), ('Market_Price', 25), ('Revenue', 67500), ('Expense', 20000), ('Profit', 47500)),
(('Farm_ID:', 5), ('Crop', 'Barley'), ('Acres_Planted', 90), ('Yield_Per_Acre', 35), ('Total_Production', 3150), ('Market_Price', 6), ('Revenue', 18900), ('Expense', 8500), ('Profit', 10400)),
(('Farm_ID', 17), ('Crop', 'Strawberries'), ('Acres_Planted', 20), ('Yield_Per_Acre', 150), ('Total_Production', 3000), ('Market_Price', 30), ('Revenue', 90000), ('Expense', 25000), ('Profit', 65000)),
(('Farm_ID', 21), ('Crop', 'Sorghum'), ('Acres_Planted', 85), ('Yield_Per_Acre', 25), ('Total_Production', 2125), ('Market_Price', 9), ('Revenue', 19125), ('Expense', 8000), ('Profit', 11125)),
(('Farm_ID:', 4), ('Crop', 'Soybeans'), ('Acres_Planted', 120), ('Yield_Per_Acre', 20), ('Total_Production', 2400), ('Market_Price', 7), ('Revenue', 16800), ('Expense', 7500), ('Profit', 9300)),
(('Farm_ID:', 1), ('Crop', 'Wheat'), ('Acres_Planted', 100), ('Yield_Per_Acre', 30), ('Total_Production', 3000), ('Market_Price', 5), ('Revenue', 15000), ('Expense', 8000), ('Profit', 7000)),
(('Farm_ID', 20), ('Crop', 'Lettuce'), ('Acres_Planted', 30), ('Yield_Per_Acre', 70), ('Total_Production', 2100), ('Market_Price', 15), ('Revenue', 31500), ('Expense', 12000), ('Profit', 19500)),
(('Farm_ID', 12), ('Crop', 'Cotton'), ('Acres_Planted', 110), ('Yield_Per_Acre', 15), ('Total_Production', 1650), ('Market_Price', 25), ('Revenue', 41250), ('Expense', 12000), ('Profit', 29250)),
(('Farm_ID', 18), ('Crop', 'Beans'), ('Acres_Planted', 55), ('Yield_Per_Acre', 40), ('Total_Production', 2200), ('Market_Price', 10), ('Revenue', 22000), ('Expense', 10000), ('Profit', 12000))}

agricultural_dataset2 ={
   (('Farm_ID', 24), ('Crop', 'Sugar Beets'), ('Acres_Planted', 75), ('Yield_Per_Acre', 40), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 14000), ('Profit', 22000)),
   (('Farm_ID', 31), ('Crop', 'Spinach'), ('Acres_Planted', 15), ('Yield_Per_Acre', 90), ('Total_Production', 1350), ('Market_Price', 18), ('Revenue', 24300), ('Expense', 10000), ('Profit', 14300)),
   (('Farm_ID', 33), ('Crop', 'Kale'), ('Acres_Planted', 35), ('Yield_Per_Acre', 80), ('Total_Production', 2800), ('Market_Price', 15), ('Revenue', 42000), ('Expense', 15000), ('Profit', 27000)),
   (('Farm_ID:', 7), ('Crop', 'Apples'), ('Acres_Planted', 30), ('Yield_Per_Acre', 100), ('Total_Production', 3000), ('Market_Price', 12), ('Revenue', 36000), ('Expense', 15000), ('Profit', 21000)),}
print(agricultural_data.symmetric_difference_update(agricultural_dataset2))
print(len(agricultural_data))
```

Ln: 38 Col: 75

Output:

```
>>>
= RESTART: D:\SMITA\MSc IT CC P1\DSPY\Programs\Agriculture_dataset_sets.py
None
30
>>>
```

Practical No. 7

Aim: Practical on functions

❖ Functions Operation on Netflix dataset:

a. mean() Code:

```
function new.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/function new.py (3.11.7)
File  Edit  Format  Run  Options  Window  Help
import pandas as pd
import random
from datetime import datetime, timedelta

# Set a seed for reproducibility
random.seed(42)

# Generate synthetic data for a smaller dataset
num_users = 30
user_ids = range(1, num_users + 1)
ages = [random.randint(18, 60) for _ in range(num_users)]
genres = ['Action', 'Comedy', 'Drama', 'Science Fiction', 'Documentary']
preferred_genres = [random.choice(genres) for _ in range(num_users)]
watch_times = [random.randint(30, 240) for _ in range(num_users)]  # Watch time in minutes

# Generate random dates for the past month
start_date = datetime.now() - timedelta(days=30)
end_date = datetime.now()
watch_dates = [start_date + timedelta(days=random.randint(0, 30)) for _ in range(num_users)]

# Create a DataFrame
data = {
    'User_ID': user_ids,
    'Age': ages,
    'Preferred_Genre': preferred_genres,
    'Watch_Time_Minutes': watch_times,
    'Watch_Date': watch_dates
}

netflix_df = pd.DataFrame(data)

# Display the smaller dataset
print("Original Dataset:")
print(netflix_df)

# Mean, Min, Max, Sum
mean_watch_time = netflix_df['Watch_Time_Minutes'].mean()
print(f"Mean Watch Time: {mean_watch_time} minutes")
```

Output:

```
IDLE Shell 3.11.7
File  Edit  Shell  Debug  Options  Window  Help
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: D:/SMITA/MSc IT CC P1/DSPY/Programs/function new.py
Original Dataset:
    User_ID  Age  ... Watch_Time_Minutes          Watch_Date
0         1   58  ...                227 2023-12-09 04:19:11.982578
1         2   25  ...                104 2023-11-30 04:19:11.982578
2         3   19  ...                 50 2023-12-02 04:19:11.982578
3         4   35  ...                 89 2023-11-27 04:19:11.982578
4         5   33  ...                 55 2023-11-17 04:19:11.982578
5         6   32  ...                127 2023-12-01 04:19:11.982578
6         7   26  ...                101 2023-11-20 04:19:11.982578
7         8   24  ...                146 2023-12-06 04:19:11.982578
8         9   52  ...                192 2023-12-04 04:19:11.982578
9        10   23  ...                123 2023-12-04 04:19:11.982578
10       11   55  ...                 71 2023-11-11 04:19:11.982578
11       12   45  ...                124 2023-11-17 04:19:11.982578
12       13   20  ...                120 2023-12-06 04:19:11.982578
13       14   19  ...                 83 2023-11-11 04:19:11.982578
14       15   23  ...                201 2023-12-05 04:19:11.982578
15       16   31  ...                 98 2023-11-20 04:19:11.982578
16       17   32  ...                209 2023-11-22 04:19:11.982578
17       18   50  ...                204 2023-11-18 04:19:11.982578
18       19   56  ...                195 2023-11-12 04:19:11.982578
19       20   19  ...                 48 2023-11-16 04:19:11.982578
20       21   53  ...                185 2023-12-09 04:19:11.982578
21       22   30  ...                192 2023-12-10 04:19:11.982578
22       23   59  ...                 73 2023-11-28 04:19:11.982578
23       24   52  ...                166 2023-12-08 04:19:11.982578
24       25   44  ...                216 2023-12-02 04:19:11.982578
25       26   32  ...                 92 2023-11-20 04:19:11.982578
26       27   46  ...                 71 2023-11-16 04:19:11.982578
27       28   55  ...                148 2023-11-30 04:19:11.982578
28       29   35  ...                127 2023-11-25 04:19:11.982578
29       30   18  ...                 99 2023-11-22 04:19:11.982578

[30 rows x 5 columns]
Mean Watch Time: 131.2 minutes
>>> 
                                                                    Ln: 40   Col: 0
```

b. min(): Code:

```
function new.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/function new.py (3.11.7)
File  Edit  Format  Run  Options  Window  Help
import pandas as pd
import random
from datetime import datetime, timedelta

# Set a seed for reproducibility
random.seed(42)

# Generate synthetic data for a smaller dataset
num_users = 30
user_ids = range(1, num_users + 1)
ages = [random.randint(18, 60) for _ in range(num_users)]
genres = ['Action', 'Comedy', 'Drama', 'Science Fiction', 'Documentary']
preferred_genres = [random.choice(genres) for _ in range(num_users)]
watch_times = [random.randint(30, 240) for _ in range(num_users)]  # Watch time in minutes

# Generate random dates for the past month
start_date = datetime.now() - timedelta(days=30)
end_date = datetime.now()
watch_dates = [start_date + timedelta(days=random.randint(0, 30)) for _ in range(num_users)]

# Create a DataFrame
data = {
    'User_ID': user_ids,
    'Age': ages,
    'Preferred_Genre': preferred_genres,
    'Watch_Time_Minutes': watch_times,
    'Watch_Date': watch_dates
}

netflix_df = pd.DataFrame(data)

# Display the smaller dataset
print("Original Dataset:")
print(netflix_df)

# Mean, Min, Max, Sum
min_watch_time = netflix_df['Watch_Time_Minutes'].min()
print(f"Minimum Watch Time: {min_watch_time} minutes")
```

Output:

```
= RESTART: D:/SMITA/MSc IT CC P1/DSPY/Programs/function new.py
Original Dataset:
   User_ID  Age  ... Watch_Time_Minutes              Watch_Date
0        1   58  ...                227 2023-12-09 04:21:01.795066
1        2   25  ...                104 2023-11-30 04:21:01.795066
2        3   19  ...                 50 2023-12-02 04:21:01.795066
3        4   35  ...                 89 2023-11-27 04:21:01.795066
4        5   33  ...                 55 2023-11-17 04:21:01.795066
5        6   32  ...                127 2023-12-01 04:21:01.795066
6        7   26  ...                101 2023-11-20 04:21:01.795066
7        8   24  ...                146 2023-12-06 04:21:01.795066
8        9   52  ...                192 2023-12-04 04:21:01.795066
9       10   23  ...                123 2023-12-04 04:21:01.795066
10      11   55  ...                 71 2023-11-11 04:21:01.795066
11      12   45  ...                124 2023-11-17 04:21:01.795066
12      13   20  ...                120 2023-12-06 04:21:01.795066
13      14   19  ...                 83 2023-11-11 04:21:01.795066
14      15   23  ...                201 2023-12-05 04:21:01.795066
15      16   31  ...                 98 2023-11-20 04:21:01.795066
16      17   32  ...                209 2023-11-22 04:21:01.795066
17      18   50  ...                204 2023-11-18 04:21:01.795066
18      19   56  ...                195 2023-11-12 04:21:01.795066
19      20   19  ...                 48 2023-11-16 04:21:01.795066
20      21   53  ...                185 2023-12-09 04:21:01.795066
21      22   30  ...                192 2023-12-10 04:21:01.795066
22      23   59  ...                 73 2023-11-28 04:21:01.795066
23      24   52  ...                166 2023-12-08 04:21:01.795066
24      25   44  ...                216 2023-12-02 04:21:01.795066
25      26   32  ...                 92 2023-11-20 04:21:01.795066
26      27   46  ...                 71 2023-11-16 04:21:01.795066
27      28   55  ...                148 2023-11-30 04:21:01.795066
28      29   35  ...                127 2023-11-25 04:21:01.795066
29      30   18  ...                 99 2023-11-22 04:21:01.795066

[30 rows x 5 columns]
Minimum Watch Time: 48 minutes
```

c. max() Code:

```
function new.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/function new.py (3.11.7)
File  Edit  Format  Run  Options  Window  Help
import pandas as pd
import random
from datetime import datetime, timedelta

# Set a seed for reproducibility
random.seed(42)

# Generate synthetic data for a smaller dataset
num_users = 30
user_ids = range(1, num_users + 1)
ages = [random.randint(18, 60) for _ in range(num_users)]
genres = ['Action', 'Comedy', 'Drama', 'Science Fiction', 'Documentary']
preferred_genres = [random.choice(genres) for _ in range(num_users)]
watch_times = [random.randint(30, 240) for _ in range(num_users)]  # Watch time in minutes

# Generate random dates for the past month
start_date = datetime.now() - timedelta(days=30)
end_date = datetime.now()
watch_dates = [start_date + timedelta(days=random.randint(0, 30)) for _ in range(num_users)]

# Create a DataFrame
data = {
    'User_ID': user_ids,
    'Age': ages,
    'Preferred_Genre': preferred_genres,
    'Watch_Time_Minutes': watch_times,
    'Watch_Date': watch_dates
}

netflix_df = pd.DataFrame(data)

# Display the smaller dataset
print("Original Dataset:")
print(netflix_df)

# Mean, Min, Max, Sum
max_watch_time = netflix_df['Watch_Time_Minutes'].max()
print(f"Maximum Watch Time: {max_watch_time} minutes")
```

Output:

```
>>>
= RESTART: D:/SMITA/MSc IT CC P1/DSPY/Programs/function new.py
Original Dataset:
   User_ID  Age  ... Watch_Time_Minutes              Watch_Date
0        1   58  ...                227 2023-12-09 04:23:41.565251
1        2   25  ...                104 2023-11-30 04:23:41.565251
2        3   19  ...                 50 2023-12-02 04:23:41.565251
3        4   35  ...                 89 2023-11-27 04:23:41.565251
4        5   33  ...                 55 2023-11-17 04:23:41.565251
5        6   32  ...                127 2023-12-01 04:23:41.565251
6        7   26  ...                101 2023-11-20 04:23:41.565251
7        8   24  ...                146 2023-12-06 04:23:41.565251
8        9   52  ...                192 2023-12-04 04:23:41.565251
9       10   23  ...                123 2023-12-04 04:23:41.565251
10      11   55  ...                 71 2023-11-11 04:23:41.565251
11      12   45  ...                124 2023-11-17 04:23:41.565251
12      13   20  ...                120 2023-12-06 04:23:41.565251
13      14   19  ...                 83 2023-11-11 04:23:41.565251
14      15   23  ...                201 2023-12-05 04:23:41.565251
15      16   31  ...                 98 2023-11-20 04:23:41.565251
16      17   32  ...                209 2023-11-22 04:23:41.565251
17      18   50  ...                204 2023-11-18 04:23:41.565251
18      19   56  ...                195 2023-11-12 04:23:41.565251
19      20   19  ...                 48 2023-11-16 04:23:41.565251
20      21   53  ...                185 2023-12-09 04:23:41.565251
21      22   30  ...                192 2023-12-10 04:23:41.565251
22      23   59  ...                 73 2023-11-28 04:23:41.565251
23      24   52  ...                166 2023-12-08 04:23:41.565251
24      25   44  ...                216 2023-12-02 04:23:41.565251
25      26   32  ...                 92 2023-11-20 04:23:41.565251
26      27   46  ...                 71 2023-11-16 04:23:41.565251
27      28   55  ...                148 2023-11-30 04:23:41.565251
28      29   35  ...                127 2023-11-25 04:23:41.565251
29      30   18  ...                 99 2023-11-22 04:23:41.565251

[30 rows x 5 columns]
Maximum Watch Time: 227 minutes
```

d. sum() Code:

```
function new.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/function new.py (3.11.7)
File  Edit  Format  Run  Options  Window  Help
import pandas as pd
import random
from datetime import datetime, timedelta

# Set a seed for reproducibility
random.seed(42)

# Generate synthetic data for a smaller dataset
num_users = 30
user_ids = range(1, num_users + 1)
ages = [random.randint(18, 60) for _ in range(num_users)]
genres = ['Action', 'Comedy', 'Drama', 'Science Fiction', 'Documentary']
preferred_genres = [random.choice(genres) for _ in range(num_users)]
watch_times = [random.randint(30, 240) for _ in range(num_users)]  # Watch time in minutes

# Generate random dates for the past month
start_date = datetime.now() - timedelta(days=30)
end_date = datetime.now()
watch_dates = [start_date + timedelta(days=random.randint(0, 30)) for _ in range(num_users)]

# Create a DataFrame
data = {
    'User_ID': user_ids,
    'Age': ages,
    'Preferred_Genre': preferred_genres,
    'Watch_Time_Minutes': watch_times,
    'Watch_Date': watch_dates
}

netflix_df = pd.DataFrame(data)

# Display the smaller dataset
print("Original Dataset:")
print(netflix_df)

# Mean, Min, Max, Sum
total_watch_time = netflix_df['Watch_Time_Minutes'].sum()
print(f"Total Watch Time: {total_watch_time} minutes")
                                                                    Ln: 38  Col: 54
```

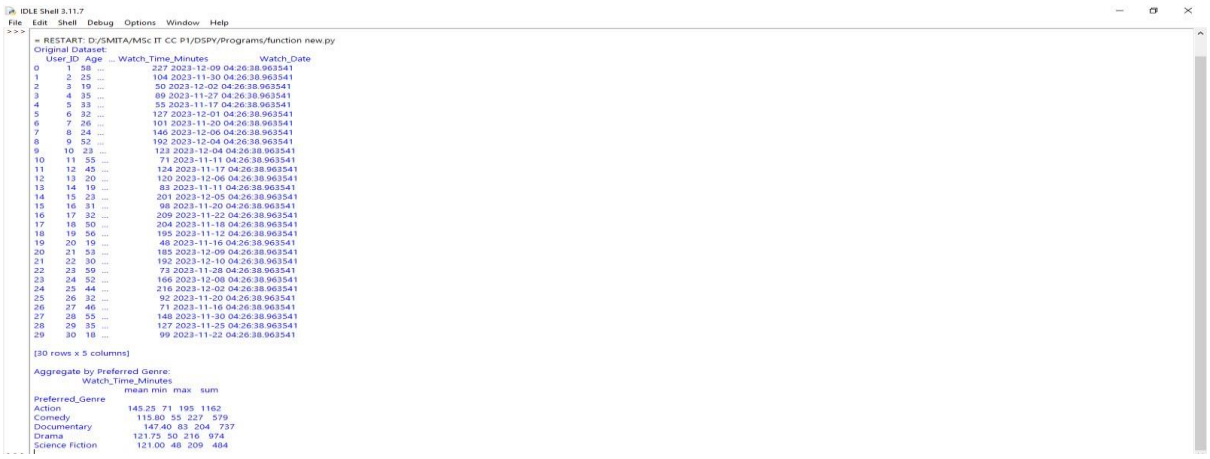
Output:

```
= RESTART: D:/SMITA/MSc IT CC P1/DSPY/Programs/function new.py
Original Dataset:
    User_ID  Age  ... Watch_Time_Minutes          Watch_Date
0         1   58  ...                227 2023-12-09 04:25:30.024008
1         2   25  ...                104 2023-11-30 04:25:30.024008
2         3   19  ...                 50 2023-12-02 04:25:30.024008
3         4   35  ...                 89 2023-11-27 04:25:30.024008
4         5   33  ...                 55 2023-11-17 04:25:30.024008
5         6   32  ...                127 2023-12-01 04:25:30.024008
6         7   26  ...                101 2023-11-20 04:25:30.024008
7         8   24  ...                146 2023-12-06 04:25:30.024008
8         9   52  ...                192 2023-12-04 04:25:30.024008
9        10   23  ...                123 2023-12-04 04:25:30.024008
10       11   55  ...                 71 2023-11-11 04:25:30.024008
11       12   45  ...                124 2023-11-17 04:25:30.024008
12       13   20  ...                120 2023-12-06 04:25:30.024008
13       14   19  ...                 83 2023-11-11 04:25:30.024008
14       15   23  ...                201 2023-12-05 04:25:30.024008
15       16   31  ...                 98 2023-11-20 04:25:30.024008
16       17   32  ...                209 2023-11-22 04:25:30.024008
17       18   50  ...                204 2023-11-18 04:25:30.024008
18       19   56  ...                195 2023-11-12 04:25:30.024008
19       20   19  ...                 48 2023-11-16 04:25:30.024008
20       21   53  ...                185 2023-12-09 04:25:30.024008
21       22   30  ...                192 2023-12-10 04:25:30.024008
22       23   59  ...                 73 2023-11-28 04:25:30.024008
23       24   52  ...                166 2023-12-08 04:25:30.024008
24       25   44  ...                216 2023-12-02 04:25:30.024008
25       26   32  ...                 92 2023-11-20 04:25:30.024008
26       27   46  ...                 71 2023-11-16 04:25:30.024008
27       28   55  ...                148 2023-11-30 04:25:30.024008
28       29   35  ...                127 2023-11-25 04:25:30.024008
29       30   18  ...                 99 2023-11-22 04:25:30.024008

[30 rows x 5 columns]
Total Watch Time: 3936 minutes
```

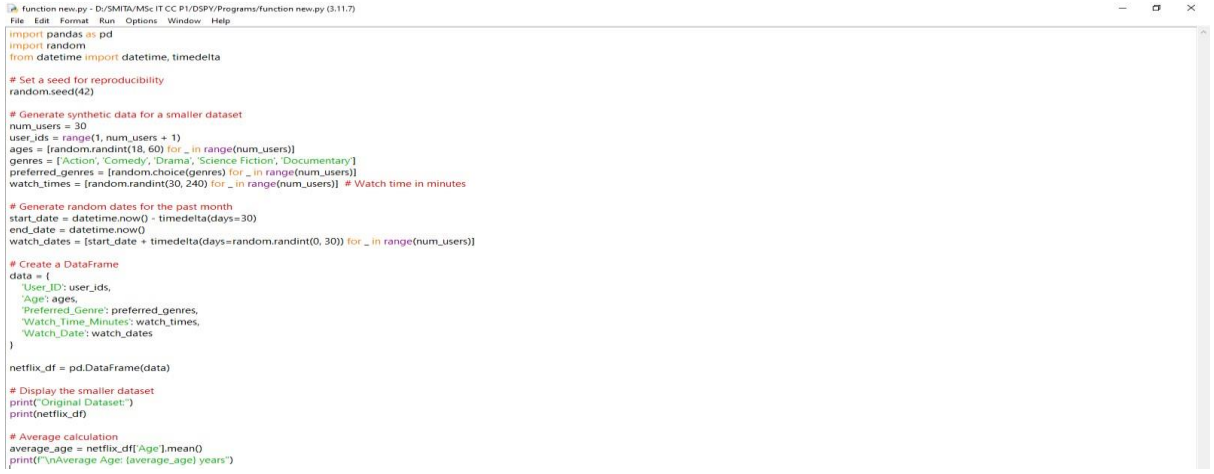
e. agg() – aggregate Code:

```
function new.py - D:/SMITA/MSc IT CC P1/DSPY/Programs/function new.py (3.11.7)
File  Edit  Format  Run  Options  Window  Help
import pandas as pd
import random
from datetime import datetime, timedelta

# Set a seed for reproducibility
random.seed(42)

# Generate synthetic data for a smaller dataset
num_users = 30
user_ids = range(1, num_users + 1)
ages = [random.randint(18, 60) for _ in range(num_users)]
genres = ['Action', 'Comedy', 'Drama', 'Science Fiction', 'Documentary']
preferred_genres = [random.choice(genres) for _ in range(num_users)]
watch_times = [random.randint(30, 240) for _ in range(num_users)]  # Watch time in minutes

# Generate random dates for the past month
start_date = datetime.now() - timedelta(days=30)
end_date = datetime.now()
watch_dates = [start_date + timedelta(days=random.randint(0, 30)) for _ in range(num_users)]

# Create a DataFrame
data = {
    'User_ID': user_ids,
    'Age': ages,
    'Preferred_Genre': preferred_genres,
    'Watch_Time_Minutes': watch_times,
    'Watch_Date': watch_dates
}

netflix_df = pd.DataFrame(data)

# Display the smaller dataset
print("Original Dataset:")
print(netflix_df)

# Aggregate function
aggregated_data = netflix_df.groupby('Preferred_Genre').agg({'Watch_Time_Minutes': ['mean', 'min', 'max', 'sum']})
print("\n\nAggregate by Preferred Genre:")
print(aggregated_data)
```

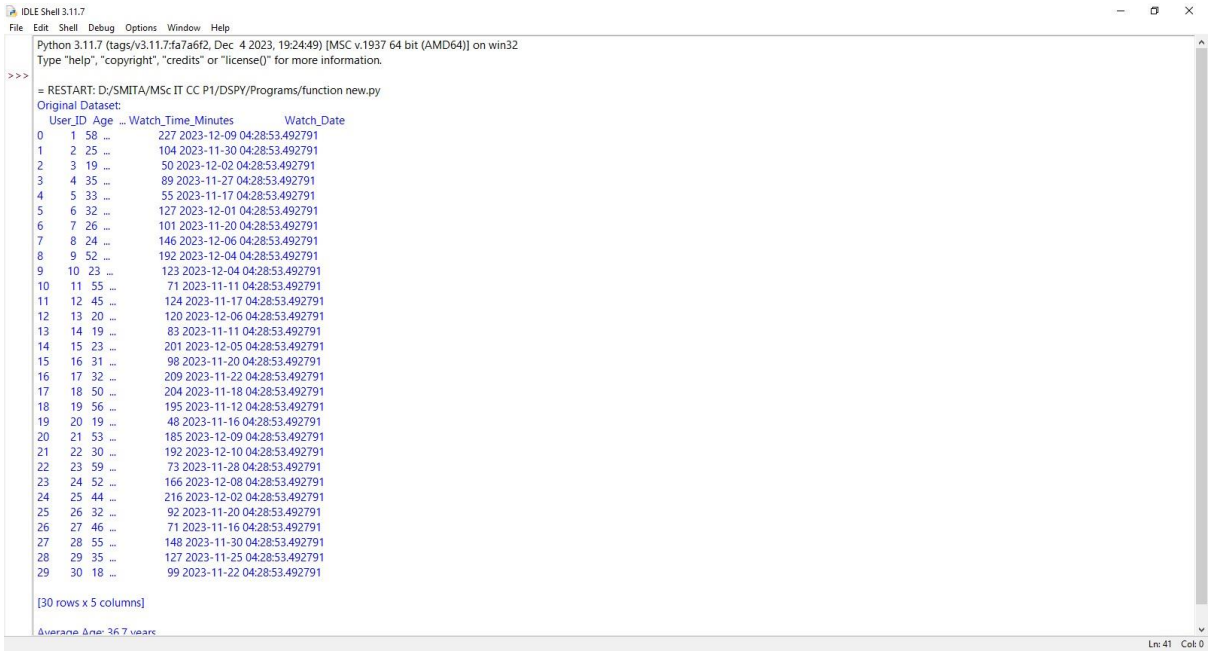
Output:



f. Average Code:



Output:



Practical 8 Aim:

Practical on exception handling and file handling

- **File Handling Operation in Python:**
  1) **Open file:**
     Code:

```
file handling.py - D:\SMITA\MSc IT CC P1\DSPY\Programs\file handling.py (3.11.7)
File  Edit  Format  Run  Options  Window  Help
f = open("my_file.txt")
f.close()
```

2) **Read:** Code:

```
*file handling.py - D:\SMITA\MSc IT CC P1\DSPY\Programs\file handling.py (3.11.7)*
File  Edit  Format  Run  Options  Window  Help
# open file in current directory
file1 = open("example.txt", "r")

# read the file
read_content = file1.read()
print(read_content)
```

Output:

```
>>> 
= RESTART: D:\SMITA\MSc IT CC P1\DSPY\Programs\file handling.py
Hello !
This is an example file.
```

3) **Write:** Code:

```
*file handling.py - D:\SMITA\MSc IT CC P1\DSPY\Programs\file handling.py (3.11.7)*
File  Edit  Format  Run  Options  Window  Help
# open file in current directory
file1 = open("example.txt", "w")
file1.write("Hello!....")
file1.close()

file1=open("example.txt", "r")
print(file1.read())
```

Output:

```
= RESTART: D:\SMITA\MSc IT CC P1\DSPY\Programs\file handling.py
Hello!....
```

```
example - Notepad
File  Edit  Format  View  Help
Hello!....
```

4) **Append:** Code:

```
file handling.py - D:\SMITA\MSc IT CC P1\DSPY\Programs\file handling.py (3.11.7)
File  Edit  Format  Run  Options  Window  Help
# open file in current directory
file1 = open("example.txt", "a")
file1.write("Now the file has more content!")
file1.close()

file1=open("example.txt", "r")
print(file1.read())
```

Output:

```
>>> 
= RESTART: D:\SMITA\MSc IT CC P1\DSPY\Programs\file handling.py
Hello!....Now the file has more content!Now the file has more content!
```

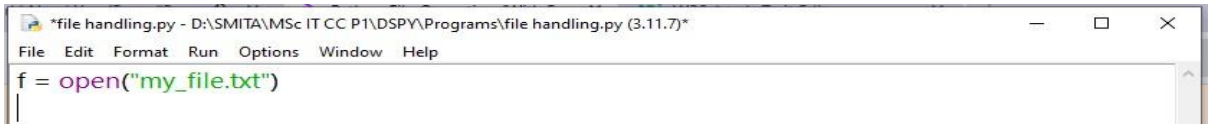
5) **Exclusive creation:** Code:

```
*file handling.py - D:\SMITA\MSc IT CC P1\DSPY\Programs\file handling.py (3.11.7)*
File  Edit  Format  Run  Options  Window  Help
f = open("my_file.txt", "x")
```

Output:

6) Close file:

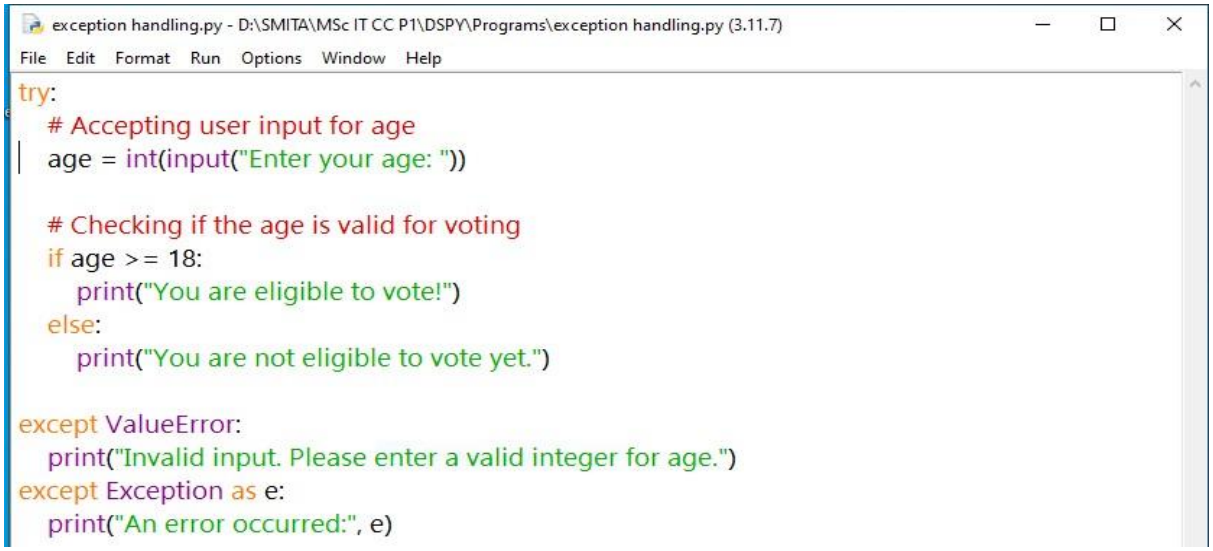
Code:



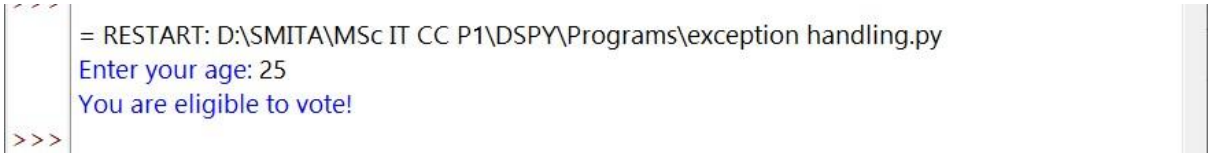
❖ Exception Handling in Python

Q.1. Accept the value from user of age to check it is valid for voting or not.

Code:

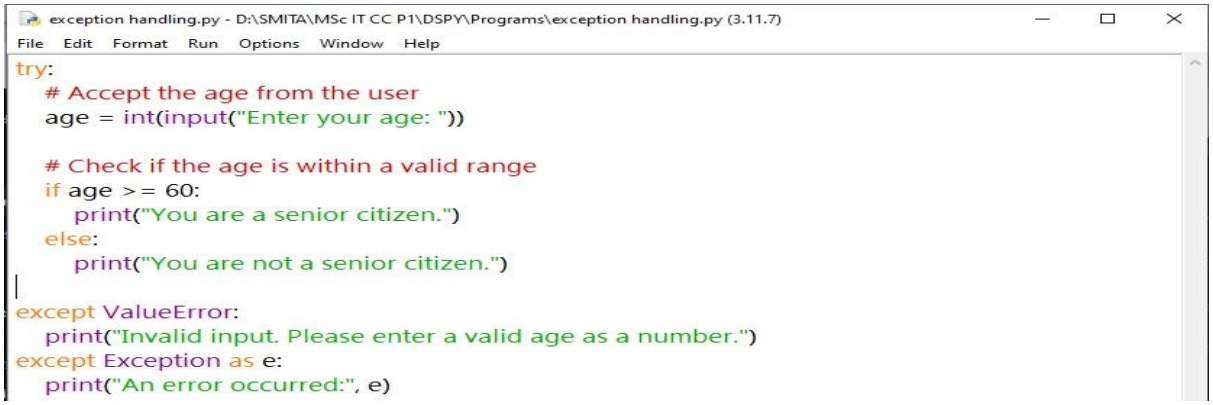


Output:



Q.2. Accept the value from user of age to check it is senior citizen or not.

Code:

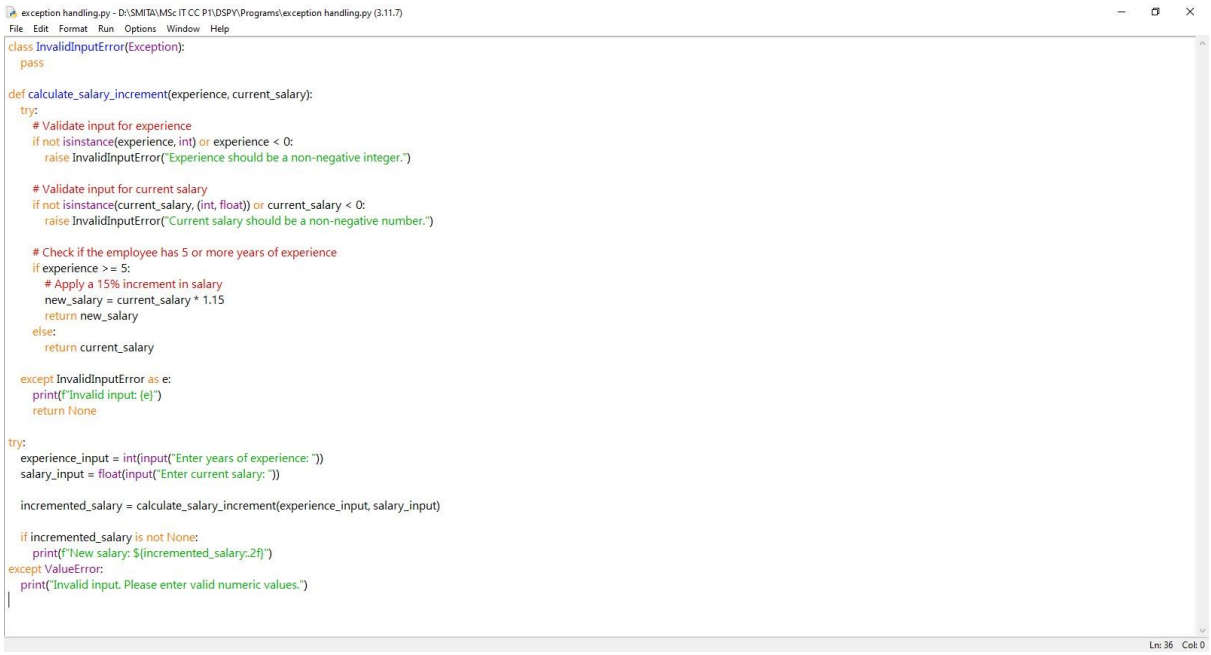


Output:

```
>>>
= RESTART: D:\SMITA\MSc IT CC P1\DSPY\Programs\exception handling.py
Enter your age: 65
You are a senior citizen.
```

Q.3. Accept experience and salary from employee if experience is 5+ then 15% increment in salary.

Code:



Output:

```
>>>
= RESTART: D:\SMITA\MSc IT CC P1\DSPY\Programs\exception handling.py
Enter years of experience: 8
Enter current salary: 75000
New salary: $86250.00
>>>
```

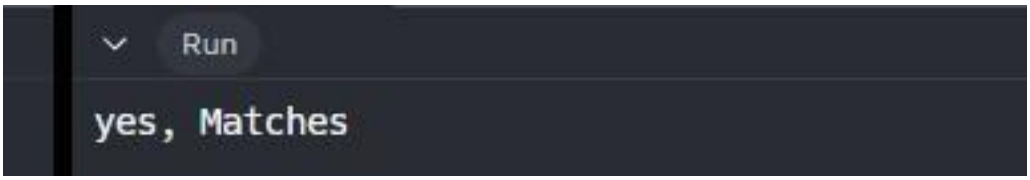
Practical 9

Aim: Practical on regular expression

1. '\': use to draw special meaning of character following it.

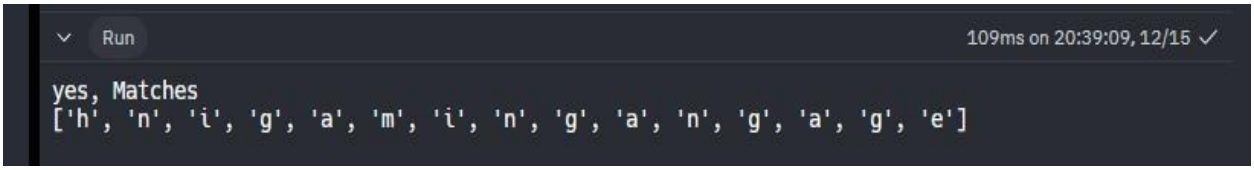
Code:

Output:



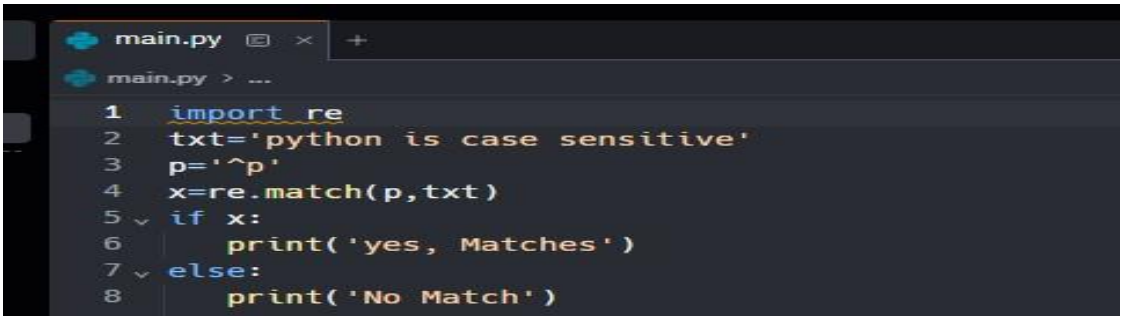
2. '[]' Returns a match for any lower-case character, alphabetically between a and n Code:

Output:

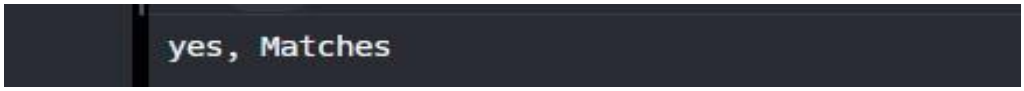


3. '^' Starts with.

Code:

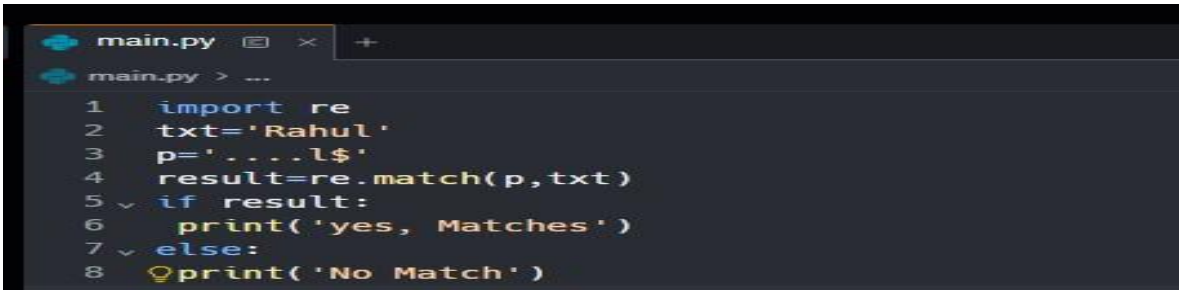


Output:



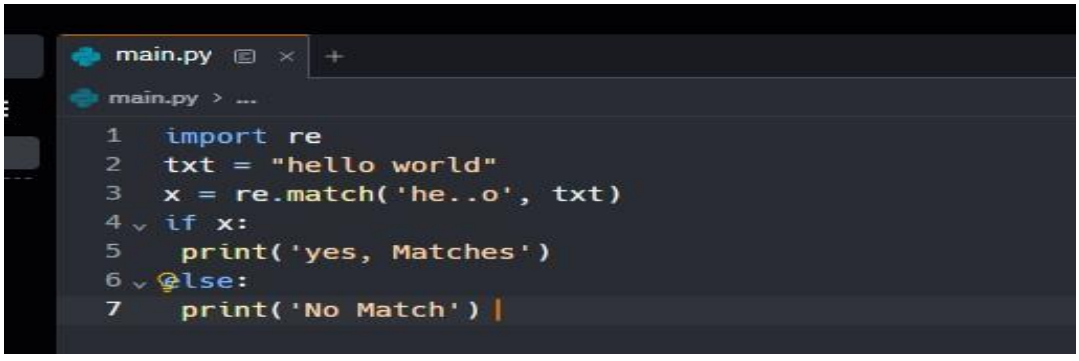
4. '$' Ends with.

Code:



Output:

5. '.' Any character (except newline character) Code:



Output:

```
Run
yes, Matches
```

6. ‘|’ Check if the string contains either "fox" or "dog":

Code:

```
main.py
main.py > ...
1  import re
2  txt = "The cat jumped over the lazy dog"
3  x = re.findall("fox|dog", txt)
4  print(x)
5  if x:
6    print("Yes, there is at least one match")
7  else:
8    print("No match")
```

Output:

```
Run
['dog']
Yes, there is at least one match
```

7. '?' Zero or one occurrences Code:

```
main.py
main.py > ...
1  import re
2  txt = "I love apple pie, but not applesauce pie."
3  "pie"
4  matches = re.findall(r'apple.?pie', txt)
5  print(matches)
```

Output:

```
Run
['apple pie']
```

8. '*' Zero or more occurrences Code:

```python
import re
txt = 'hello world'
"o"
x = re.findall("he.*o", txt)
print(x)
```

Output:

```
['hello wo']
```

9. '+' One or more occurrences Code:

```python
import re
txt = "The cat jumped over the lazy dog."
matches = re.findall(r'cat.+dog',txt)
print(matches)
```

Output:

```
Run
['cat jumped over the lazy dog']
```

10. '{}' Exactly the specified number of occurrences

Code:

```python
import re
txt = "apple pie"
x = re.findall("a.{3}e", txt)
print(x)
```

Output:

```
Run
['apple']
```

11. Enclose a group of regex Code:

```python
import re
txt = "The price of apples is $1.25 and the price of oranges is $0.75."
matches = '(oranges|apples) is \$(\d+\.\d+)'
for match in re.finditer(matches, txt):
print(f"The price of '{match.group()}'")
```

Output:

```
Run
The price of 'apples is $1.25'
The price of 'oranges is $0.75'
```

❖ **Application of Regular Expression 1.**

Web scraping and Data Collection (url)

Code:

```python
import re
import requests
# Example: Extracting URLs from the YouTube Home page
url = 'https://accounts.google.com/v3/signin/identifier?
authuser=0&continue=https%3A%2com%2Fmail%2Fdata&ec=GAlAFw&hl=en-
GB&service=mail&flowName=GlifWebSignIn&flowEntry=AddSession&5775%3A170265489076574&the
me=glif'
response = requests.get(url)
# Check if the request was successful (status code 200)
if response.status_code == 200:
 # Extracting URLs using regular expression
    html_content = response.text
    url_pattern = re.compile(r'href="(https?://.*?)"')
    urls = url_pattern.findall(html_content)
 # Displaying the extracted URLs
for extracted_url in urls:
    print(extracted_url)
else:
 print(f"Failed to retrieve the webpage. Status Code: {response.status_code}")
```

Output:

```
Run                                                    1s on 21:11:51, 12/15 ✓
https://accounts.google.com/v3/signin/
https://www.gstatic.com/_/mss/boq-identity/_/ss/k=boq-identity.AccountsSignInUi.EcigJeQdnIY.L.
W.O/am=P4GFJI4NgBgzyzl_zzg5DAAAAAAAQAAAsAawgw/d=1/ed=1/rs=AOaEmlGf6vdU1wnYZish42M5Ym3ccNcRbQ/m
=identifierview,_b,_tp
https://support.google.com/accounts?p=signin_privatebrowsing&amp;hl=en-GB
https://support.google.com/accounts?hl=en-GB&amp;p=account_iph
https://accounts.google.com/TOS?loc=IN&amp;hl=en-GB&amp;privacy=true
https://accounts.google.com/TOS?loc=IN&amp;hl=en-GB
```

2. Natural Language Processing Code:

```python
import re
text="Natural language processing is machine learning technology"
tokens = re.findall(r'\b\w+\b', text)
print(tokens)
```

Output:

```
Run                                                    141ms on 21:18:33, 12/15 ✓
['Natural', 'language', 'processing', 'is', 'machine', 'learning', 'technology']
```

3. Entity Recognition:

Code:

```python
import re
text = "Appointment on 2022-12-15 at 2:30 PM. Call 123-456-7891 for details."
dates = re.findall(r'\b\d{4}-\d{2}-\d{2}\b', text)
phone_numbers = re.findall(r'\d{3}-\d{3}-\d{4}', text)
print(dates,'\n',phone_numbers)
```

Output:

```
Run
['2022-12-15']
 ['123-456-7891']
```

3. pattern detection for iris, emails, text, date time manipulation

a) Pattern Detection in Iris Data:

Code:

```python
import re
iris_data = "SepalLength:5.1 SepalWidth:3.5 PetalLength:1.4 PetalWidth:0.2"
matches = re.findall(r'\d+\.\d+',iris_data)
print(matches)
```

Output:

```
Run
['5.1', '3.5', '1.4', '0.2']
```

b) Email Pattern Detection:

Code:

```
main.py
main.py > ...
1  import re
2  text = "Contact us at xyz@gmail.com or support-in@google.com."
3  email_pattern = re.compile(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b')
4  emails = email_pattern.findall(text)
5  print(emails)
```

Output:

```
Run
['xyz@gmail.com', 'support-in@google.com']
```

c) Text Detection:

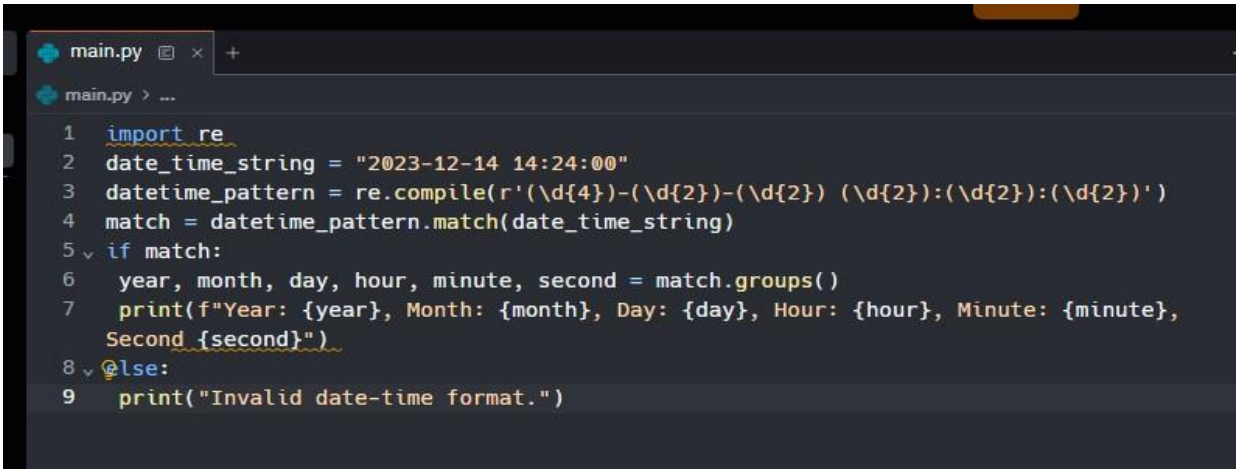
Code:

```
main.py
main.py > ...
1  import re
2  text = "The quick brown fox jumps over the lazy dog."
3  match = re.search(r'brown fox',text)
4  if match:
5   print("Pattern found:", match.group())
6  else:
7   print("Pattern not found.")
```

Output:

d) Date and Time Manipulation:

Code:



Output: