TECHNIK NEST
INNOVATIVE MINDS, NESTING SUCCESS

Name: Ymeen Fatima

Intern ID: TN_IN02_PY_033

Task no: week-task-02

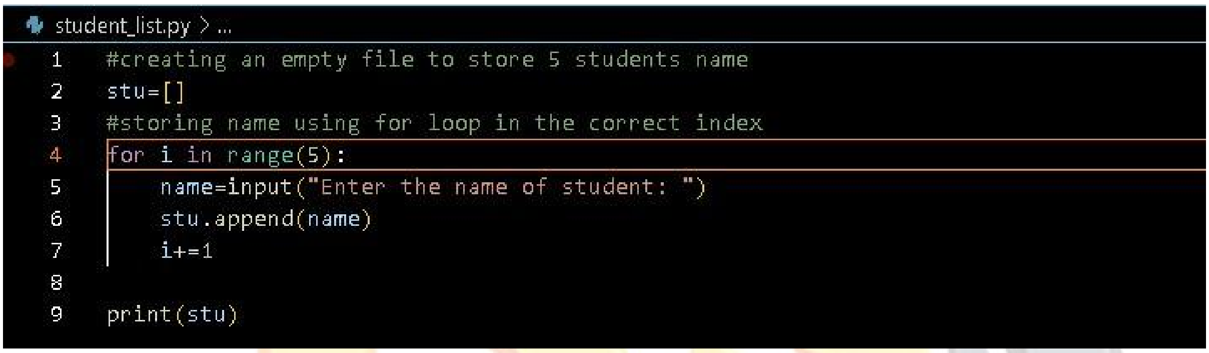
E-mail: ymeenfatima192gmail.com

Internship Domain: python Development

Task: Lists

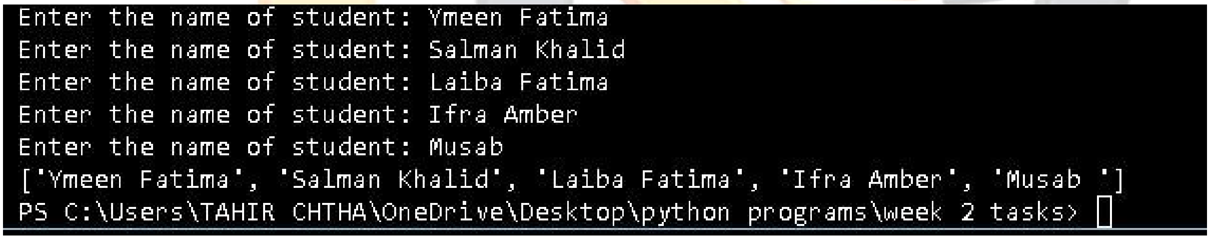
1. Store 5 student names & print each.

- **Creating a list storing 5 names by initializing empty list**
- **Applying for loop to insert name**
- **Print the output**



```python
#creating an empty file to store 5 students name
stu=[]
#storing name using for loop in the correct index
for i in range(5):
    name=input("Enter the name of student: ")
    stu.append(name)
    i+=1

print(stu)
```

Output:

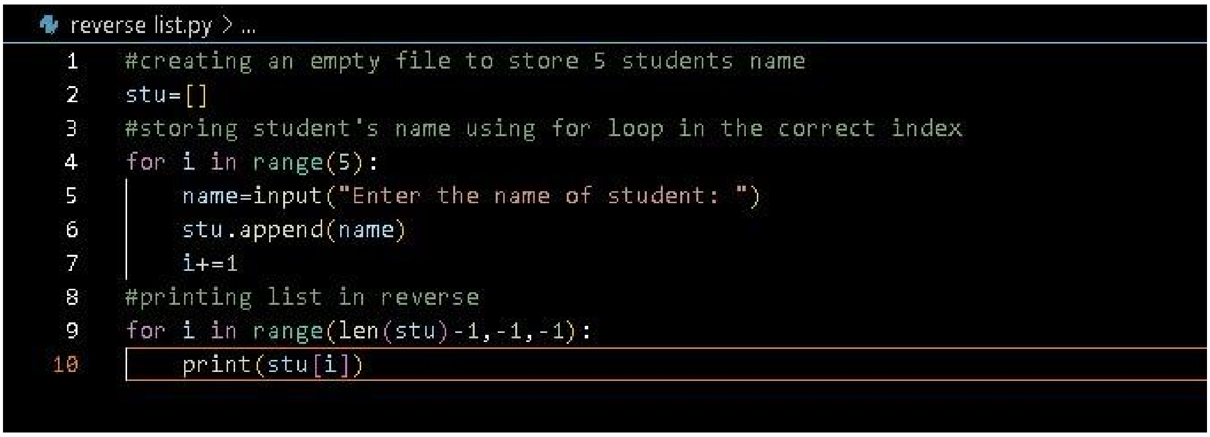


```
Enter the name of student: Ymeen Fatima
Enter the name of student: Salman Khalid
Enter the name of student: Laiba Fatima
Enter the name of student: Ifra Amber
Enter the name of student: Musab
['Ymeen Fatima', 'Salman Khalid', 'Laiba Fatima', 'Ifra Amber', 'Musab ']
PS C:\Users\TAHIR CHTHA\OneDrive\Desktop\python programs\week 2 tasks> 
```

- **Learn how to store data in an empty list and print it**

2. Reverse list without reverse().

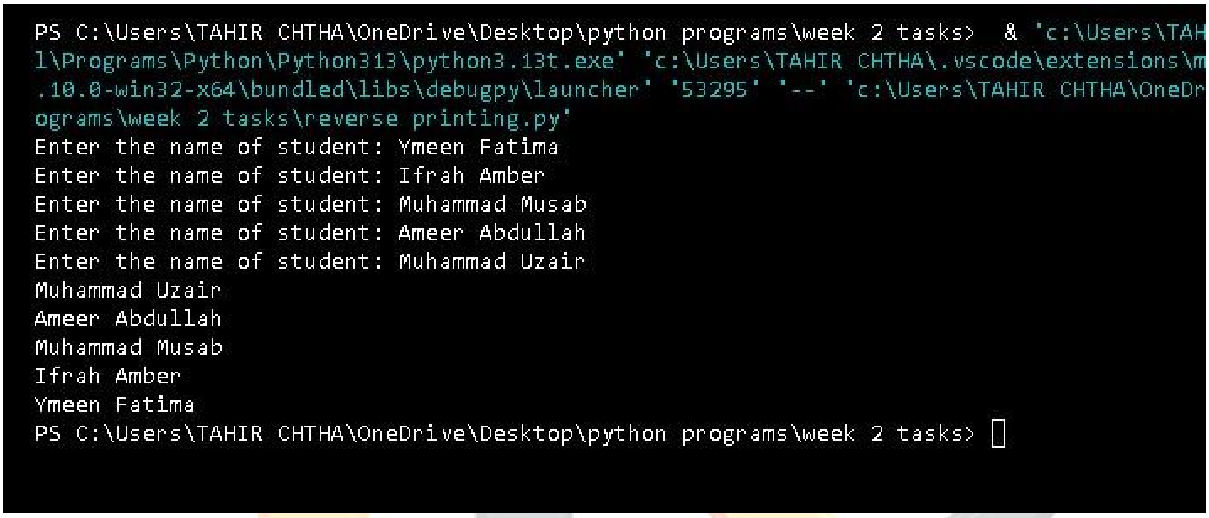
- **Make an empty file then store the data in it using for loop**
- **After storing the data use a for loop to print the entries of list in reverse order**



reverse list.py > ...

```python
#creating an empty file to store 5 students name
stu=[]
#storing student's name using for loop in the correct index
for i in range(5):
    name=input("Enter the name of student: ")
    stu.append(name)
    i+=1
#printing list in reverse
for i in range(len(stu)-1,-1,-1):
    print(stu[i])
```

Output:



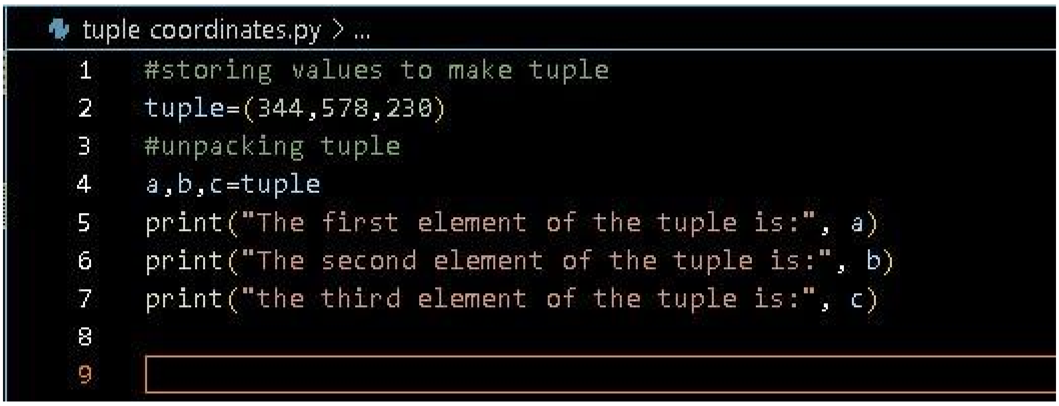
```
PS C:\Users\TAHIR CHTHA\OneDrive\Desktop\python programs\week 2 tasks>  & 'c:\Users\TAH
l\Programs\Python\Python313\python3.13t.exe' 'c:\Users\TAHIR CHTHA\.vscode\extensions\m
.10.0-win32-x64\bundled\libs\debugpy\launcher' '53295' '--' 'c:\Users\TAHIR CHTHA\OneDr
ograms\week 2 tasks\reverse printing.py'
Enter the name of student: Ymeen Fatima
Enter the name of student: Ifrah Amber
Enter the name of student: Muhammad Musab
Enter the name of student: Ameer Abdullah
Enter the name of student: Muhammad Uzair
Muhammad Uzair
Ameer Abdullah
Muhammad Musab
Ifrah Amber
Ymeen Fatima
PS C:\Users\TAHIR CHTHA\OneDrive\Desktop\python programs\week 2 tasks> 
```

- **Learn to reverse the printing of list without using bilt in function reverse().**

Tasks :Tuples

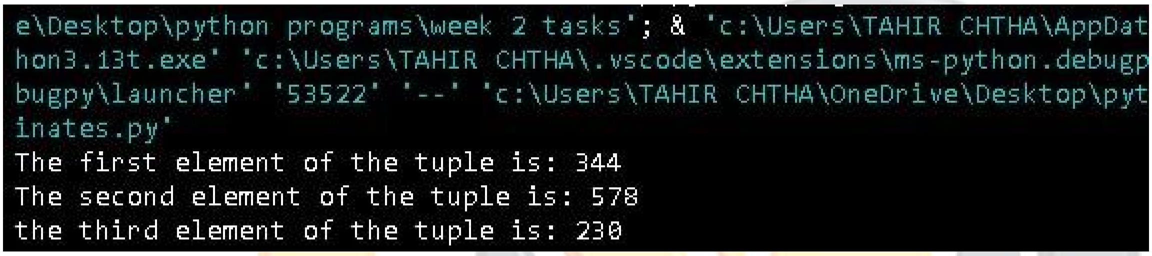
1. **Store 3 coordinates & unpack**

- **Store values in a tuple**
- **Unpacked them**
- **Print individually**



```python
#storing values to make tuple
tuple=(344,578,230)
#unpacking tuple
a,b,c=tuple
print("The first element of the tuple is:", a)
print("The second element of the tuple is:", b)
print("the third element of the tuple is:", c)

```

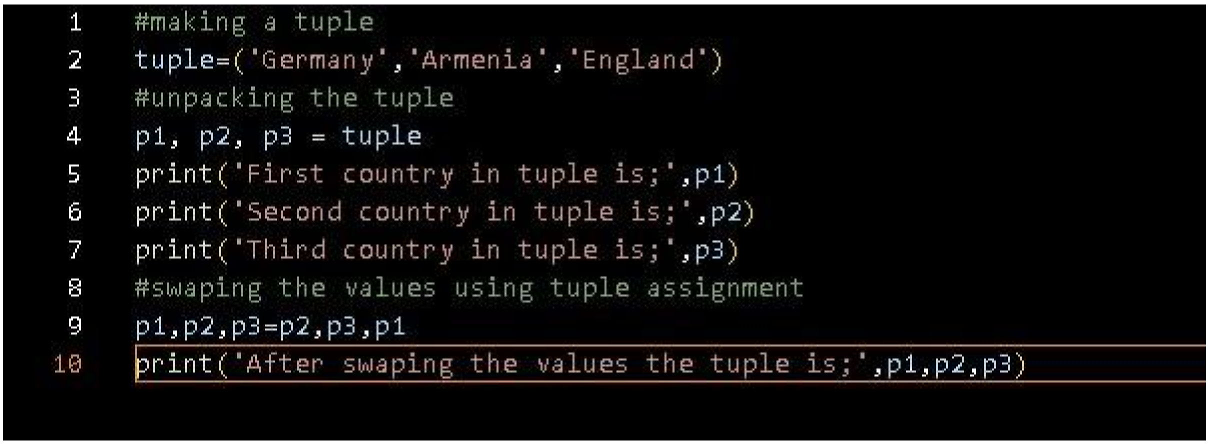
Output:



```
e\Desktop\python programs\week 2 tasks'; & 'c:\Users\TAHIR CHTHA\AppDat
hon3.13t.exe' 'c:\Users\TAHIR CHTHA\.vscode\extensions\ms-python.debugp
bugpy\launcher' '53522' '--' 'c:\Users\TAHIR CHTHA\OneDrive\Desktop\pyt
inates.py'
The first element of the tuple is: 344
The second element of the tuple is: 578
the third element of the tuple is: 230
```

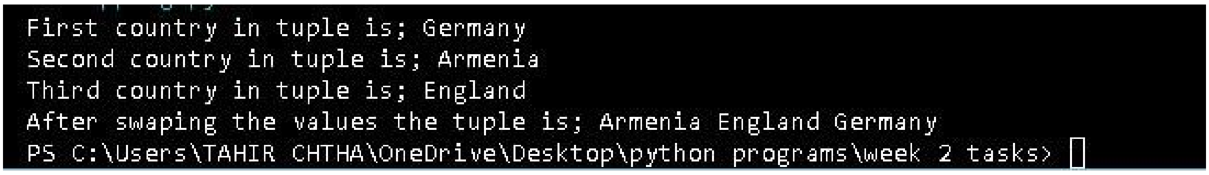
2. Swap two values of tuple using tuple assignment

- Store values in tuple
- Unpacked them
- Swap them using tuple assignment
- Print the result individually



```python
#making a tuple
tuple=('Germany','Armenia','England')
#unpacking the tuple
p1, p2, p3 = tuple
print('First country in tuple is;',p1)
print('Second country in tuple is;',p2)
print('Third country in tuple is;',p3)
#swaping the values using tuple assignment
p1,p2,p3=p2,p3,p1
print('After swaping the values the tuple is;',p1,p2,p3)
```

Output:

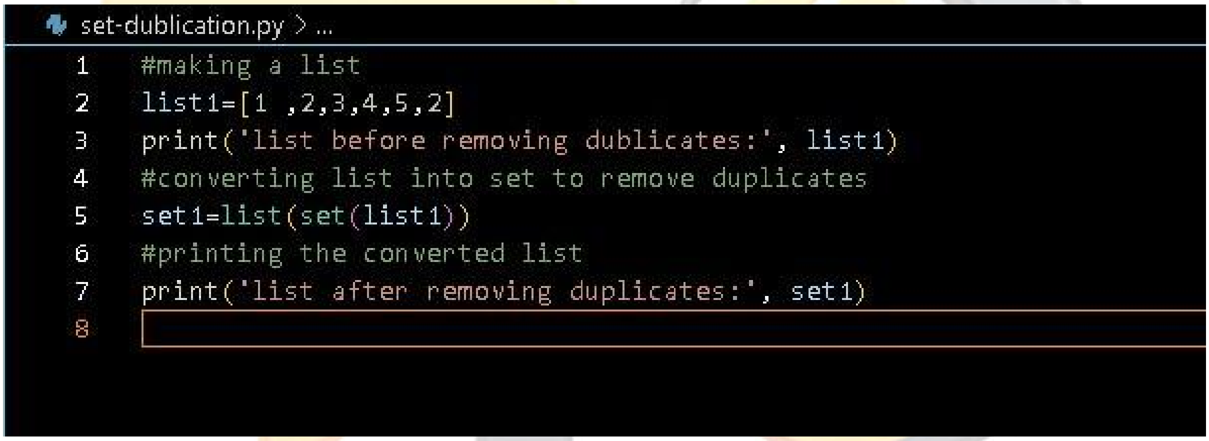


```
First country in tuple is; Germany
Second country in tuple is; Armenia
Third country in tuple is; England
After swaping the values the tuple is; Armenia England Germany
PS C:\Users\TAHIR CHTHA\OneDrive\Desktop\python programs\week 2 tasks> 
```

Tasks – Sets

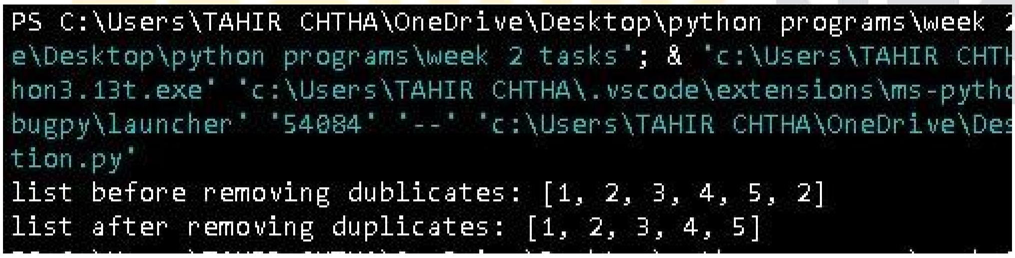
1.Remove duplicates from list

- Making a list
- Convert into set
- Then print it the duplicates will remove automatically



```python
#making a list
list1=[1 ,2,3,4,5,2]
print('list before removing dublicates:', list1)
#converting list into set to remove duplicates
set1=list(set(list1))
#printing the converted list
print('list after removing duplicates:', set1)

```

Output:



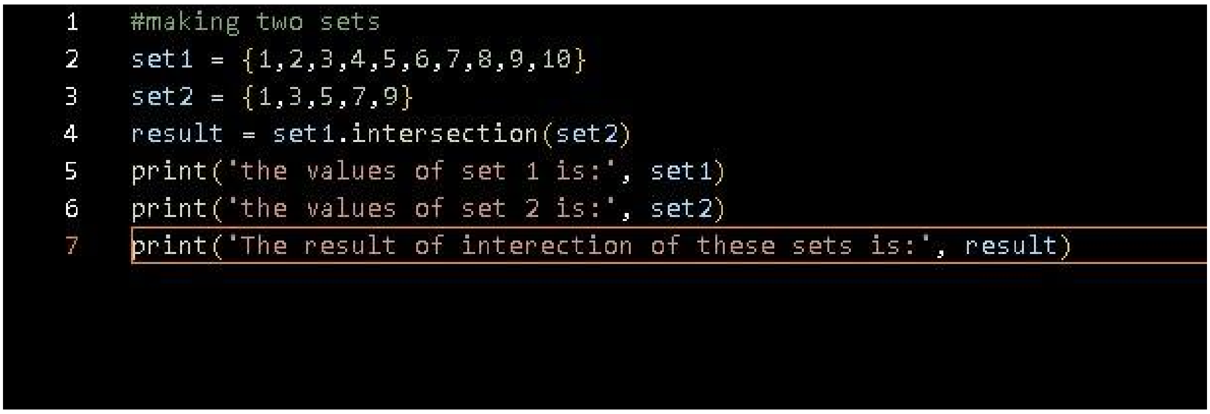
```
PS C:\Users\TAHIR CHTHA\OneDrive\Desktop\python programs\week 2
e\Desktop\python programs\week 2 tasks'; & 'c:\Users\TAHIR CHTH
hon3.13t.exe' 'c:\Users\TAHIR CHTHA\.vscode\extensions\ms-pytho
bugpy\launcher' '54084' '--' 'c:\Users\TAHIR CHTHA\OneDrive\Des
tion.py'
list before removing dublicates: [1, 2, 3, 4, 5, 2]
list after removing duplicates: [1, 2, 3, 4, 5]
```

- Learn how to convert list into set

3. Find intersection of two sets:

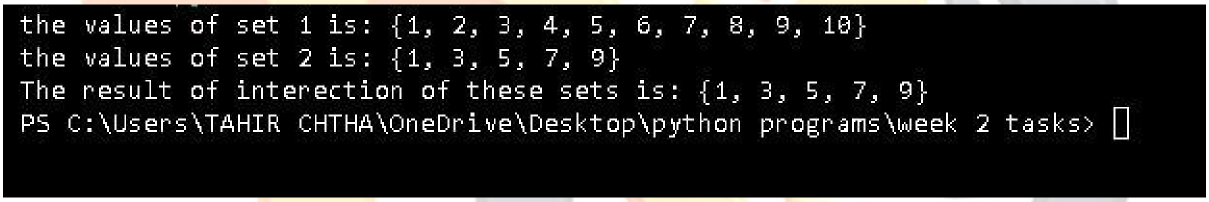
- Making 2 sets

- Using intersection built-in function to calculate intersection of 2 sets and print the result.



```
#making two sets
set1 = {1,2,3,4,5,6,7,8,9,10}
set2 = {1,3,5,7,9}
result = set1.intersection(set2)
print('the values of set 1 is:', set1)
print('the values of set 2 is:', set2)
print('The result of interection of these sets is:', result)
```

Output:



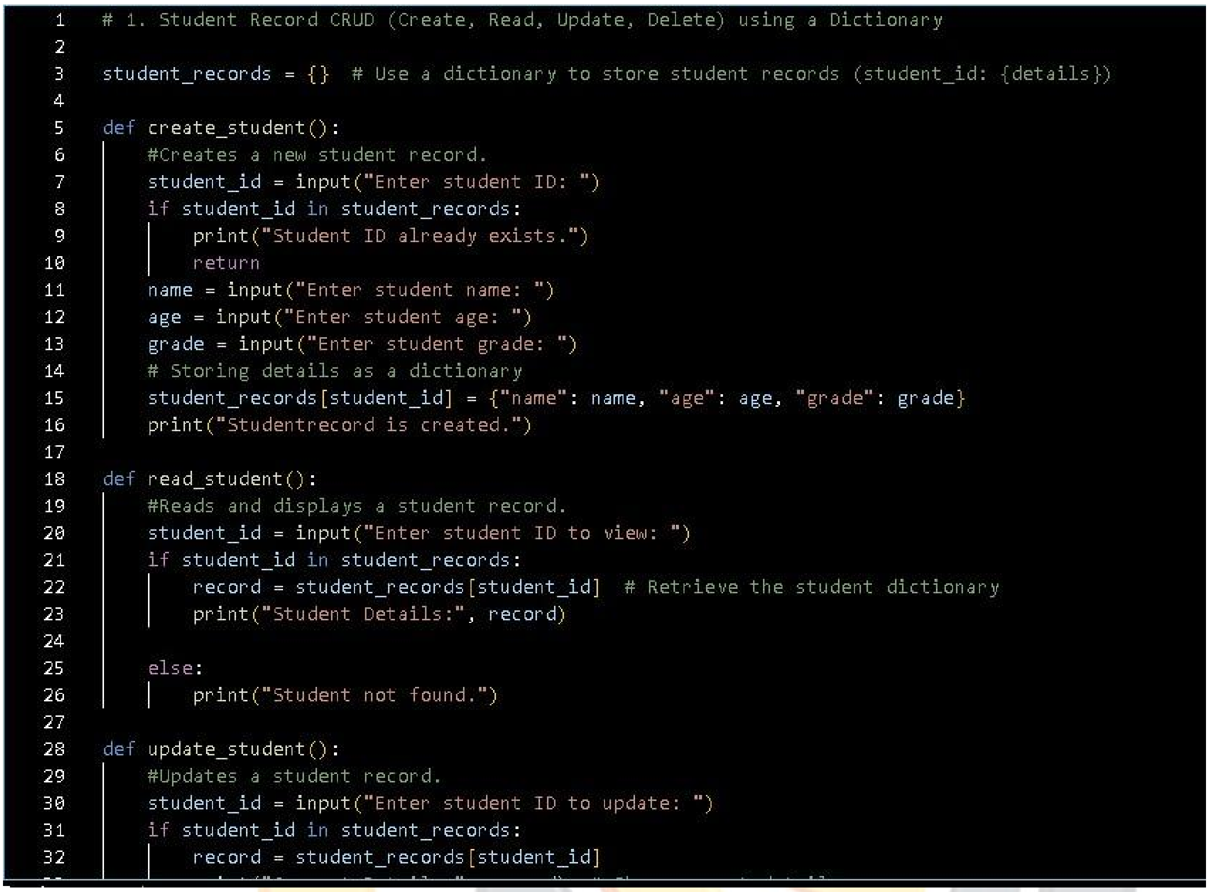
```
the values of set 1 is: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
the values of set 2 is: {1, 3, 5, 7, 9}
The result of interection of these sets is: {1, 3, 5, 7, 9}
PS C:\Users\TAHIR CHTHA\OneDrive\Desktop\python programs\week 2 tasks> 
```

Learn how to use built-in function

Tasks – Dictionaries

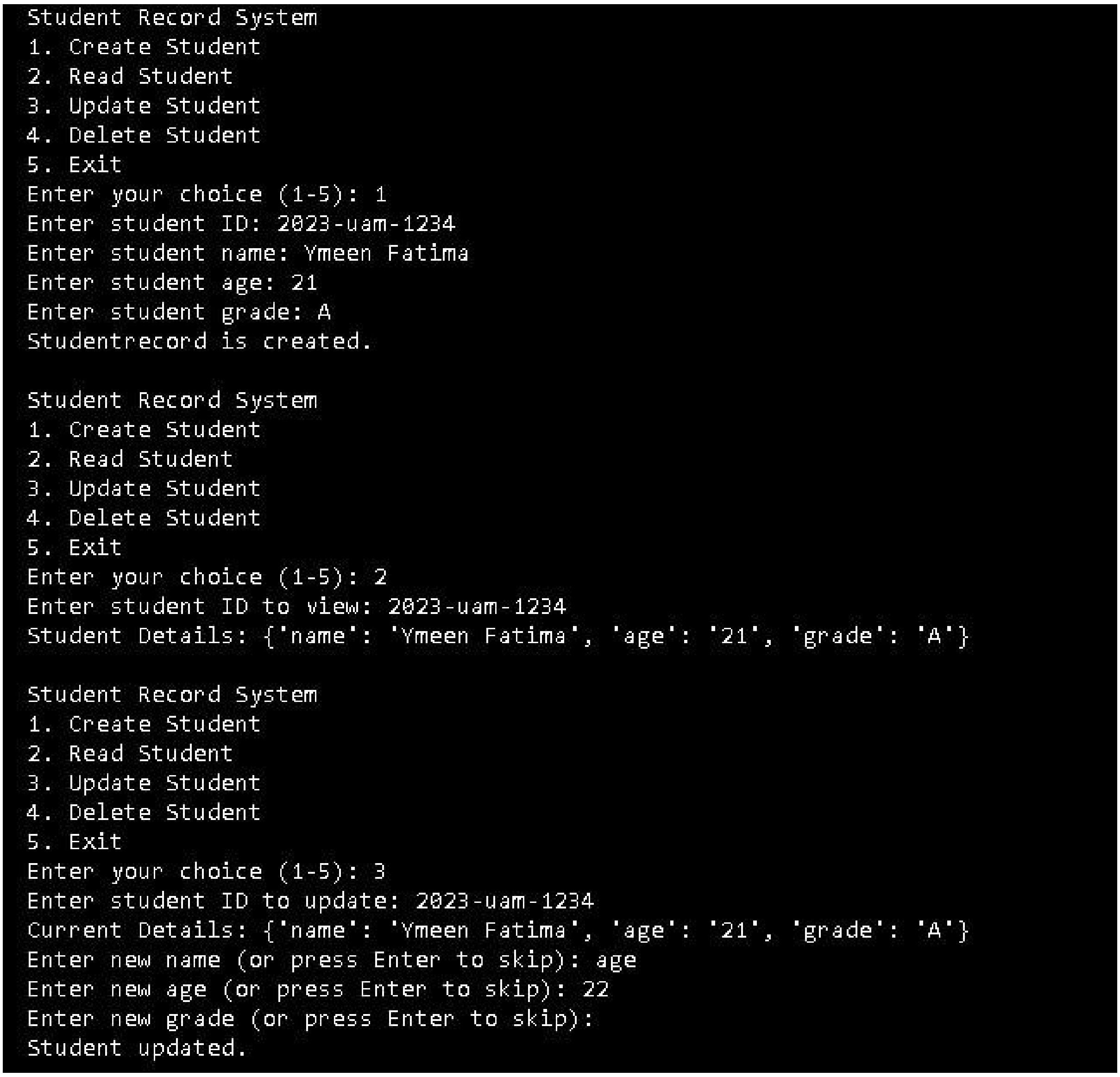
1. Student record CRUD in dict.:.

- Initialize an empty dictionary
- Make functions to create, read, update and delete the student record that will store in the dictiionary
- Print output



```python
# 1. Student Record CRUD (Create, Read, Update, Delete) using a Dictionary

student_records = {}  # Use a dictionary to store student records (student_id: {details})

def create_student():
    #Creates a new student record.
    student_id = input("Enter student ID: ")
    if student_id in student_records:
        print("Student ID already exists.")
        return
    name = input("Enter student name: ")
    age = input("Enter student age: ")
    grade = input("Enter student grade: ")
    # Storing details as a dictionary
    student_records[student_id] = {"name": name, "age": age, "grade": grade}
    print("Studentrecord is created.")

def read_student():
    #Reads and displays a student record.
    student_id = input("Enter student ID to view: ")
    if student_id in student_records:
        record = student_records[student_id]  # Retrieve the student dictionary
        print("Student Details:", record)

    else:
        print("Student not found.")

def update_student():
    #Updates a student record.
    student_id = input("Enter student ID to update: ")
    if student_id in student_records:
        record = student_records[student_id]
```

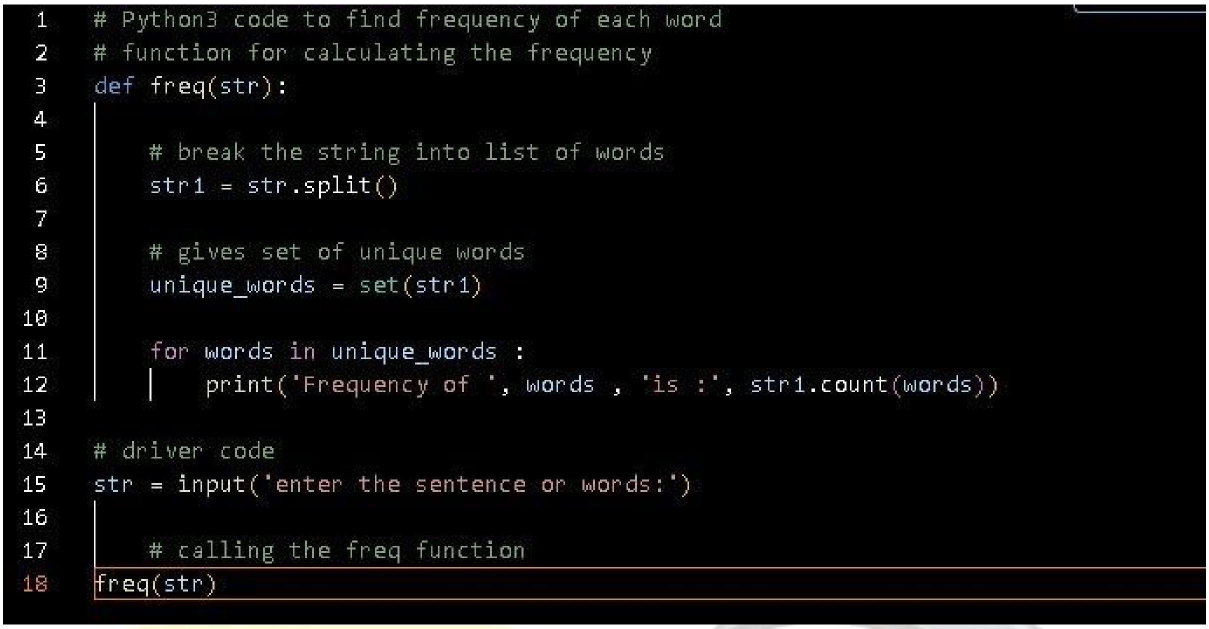
Output:



```
Student Record System
1. Create Student
2. Read Student
3. Update Student
4. Delete Student
5. Exit
Enter your choice (1-5): 1
Enter student ID: 2023-uam-1234
Enter student name: Ymeen Fatima
Enter student age: 21
Enter student grade: A
Studentrecord is created.

Student Record System
1. Create Student
2. Read Student
3. Update Student
4. Delete Student
5. Exit
Enter your choice (1-5): 2
Enter student ID to view: 2023-uam-1234
Student Details: {'name': 'Ymeen Fatima', 'age': '21', 'grade': 'A'}

Student Record System
1. Create Student
2. Read Student
3. Update Student
4. Delete Student
5. Exit
Enter your choice (1-5): 3
Enter student ID to update: 2023-uam-1234
Current Details: {'name': 'Ymeen Fatima', 'age': '21', 'grade': 'A'}
Enter new name (or press Enter to skip): age
Enter new age (or press Enter to skip): 22
Enter new grade (or press Enter to skip):
Student updated.
```

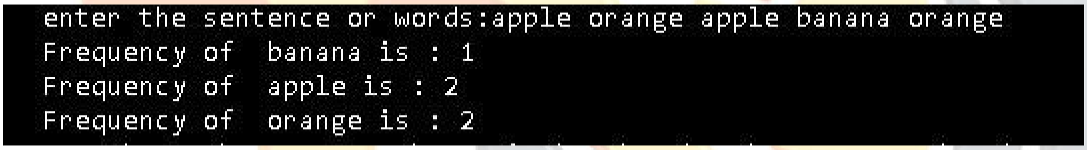
2.Count word frequency in sentence.

- Take sentence or multiple words from the user
- Define a function to count the frequency of words
- Passing parameter str in the function
- Split the words
- Count them using count function
- And display result



```python
# Python3 code to find frequency of each word
# function for calculating the frequency
def freq(str):

    # break the string into list of words
    str1 = str.split()

    # gives set of unique words
    unique_words = set(str1)

    for words in unique_words :
        print('Frequency of ', words , 'is :', str1.count(words))

# driver code
str = input('enter the sentence or words:')

    # calling the freq function
freq(str)
```

Output:

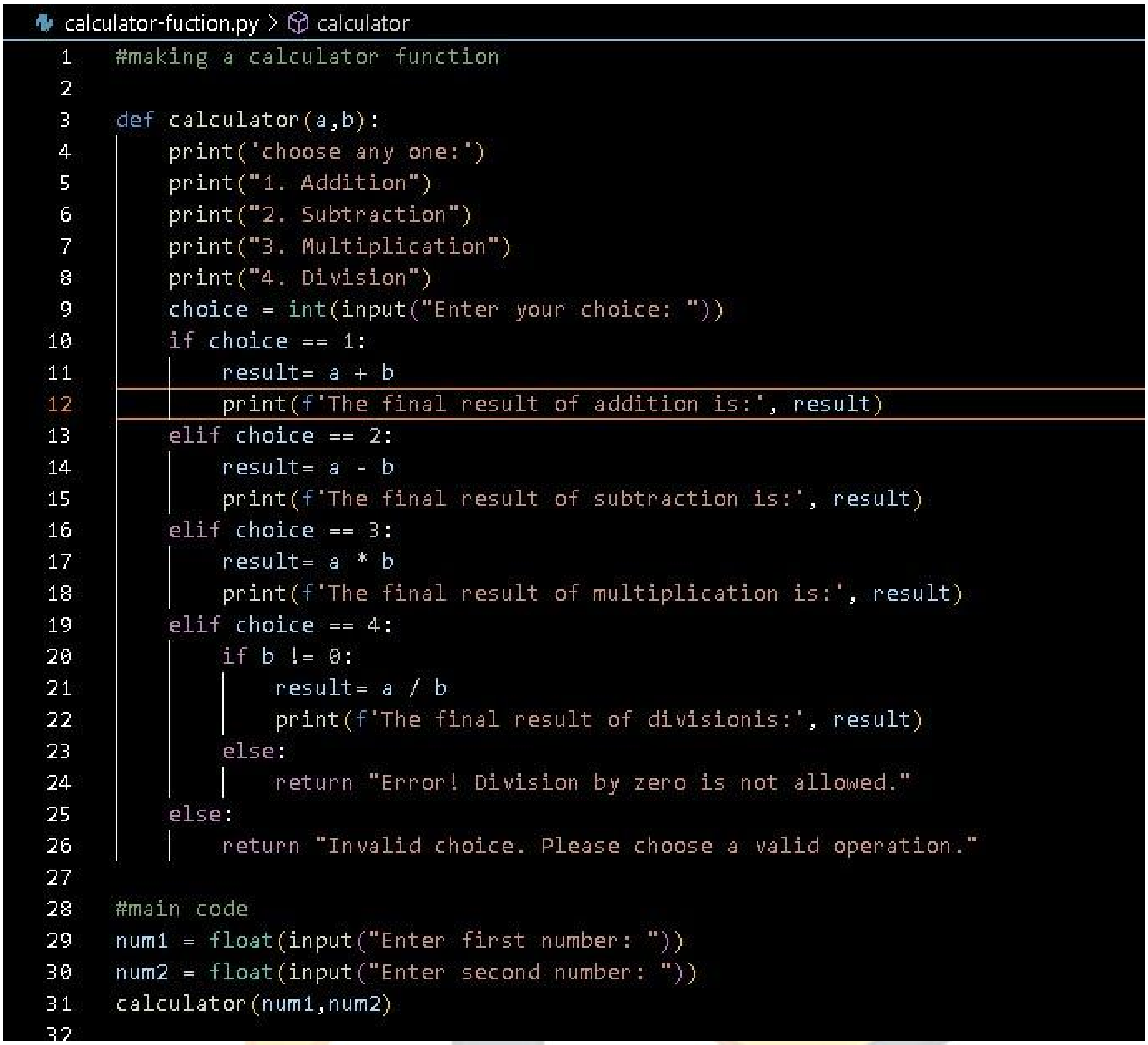


```
enter the sentence or words:apple orange apple banana orange
Frequency of  banana is : 1
Frequency of  apple is : 2
Frequency of  orange is : 2
```

Tasks – Functions

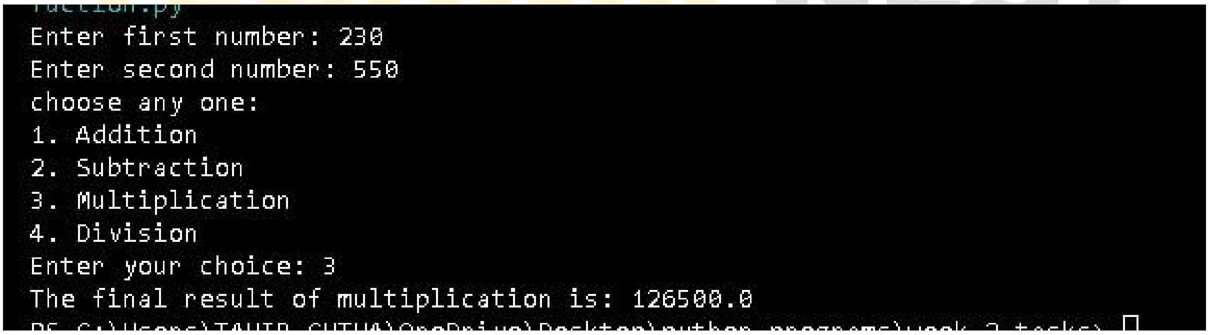
1. Write calc(a,b,op).

- Take input from user and ask about desired operation
- Pass the parameter to perform the the user desired operation
- The operation will be performed on the perameter and return the result
- Display the result



```python
#making a calculator function

def calculator(a,b):
    print('choose any one:')
    print("1. Addition")
    print("2. Subtraction")
    print("3. Multiplication")
    print("4. Division")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        result= a + b
        print(f'The final result of addition is:', result)
    elif choice == 2:
        result= a - b
        print(f'The final result of subtraction is:', result)
    elif choice == 3:
        result= a * b
        print(f'The final result of multiplication is:', result)
    elif choice == 4:
        if b != 0:
            result= a / b
            print(f'The final result of divisionis:', result)
        else:
            return "Error! Division by zero is not allowed."
    else:
        return "Invalid choice. Please choose a valid operation."

#main code
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
calculator(num1,num2)
```

Output:

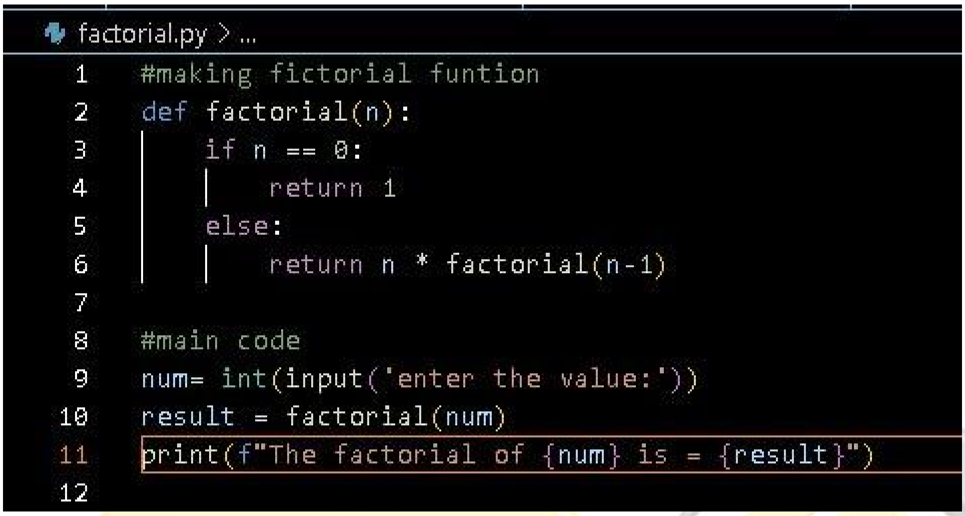


```
fuction.py
Enter first number: 230
Enter second number: 550
choose any one:
1. Addition
2. Subtraction
3. Multiplication
4. Division
Enter your choice: 3
The final result of multiplication is: 126500.0
```

2. Write factorial(n) recursive.

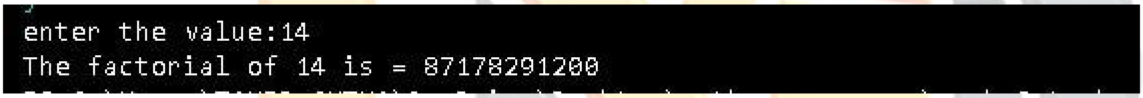
- This program finds the factorial of the number given by the user
- Factorial function is defined

- If n=0 it will return 1
- Or else it will return the factorial of the number.



```python
#making fictorial funtion
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

#main code
num= int(input('enter the value:'))
result = factorial(num)
print(f"The factorial of {num} is = {result}")
```

Output:

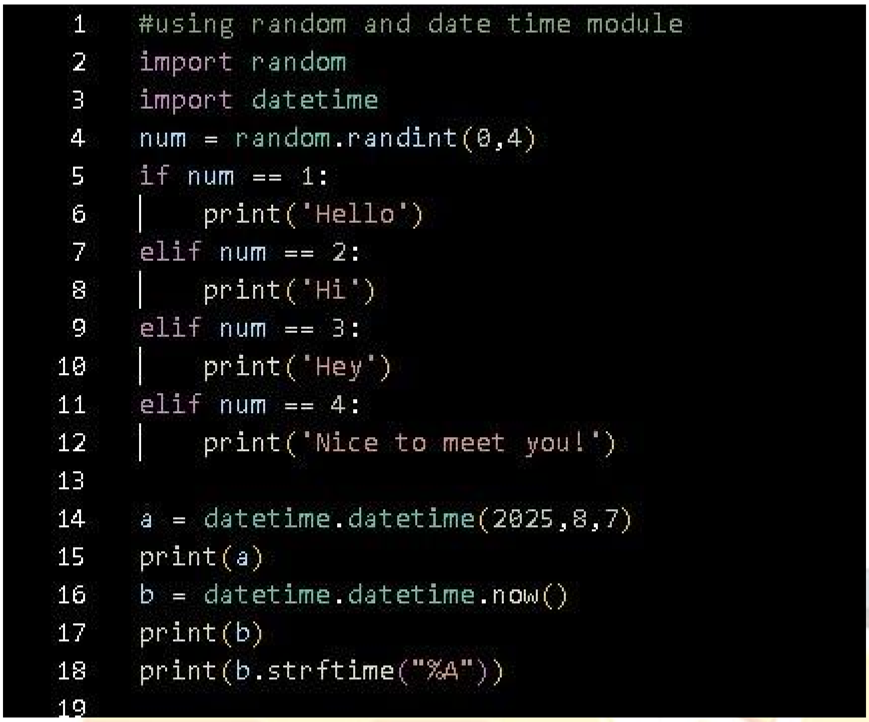


```
enter the value:14
The factorial of 14 is = 87178291200
```

Tasks –Modules

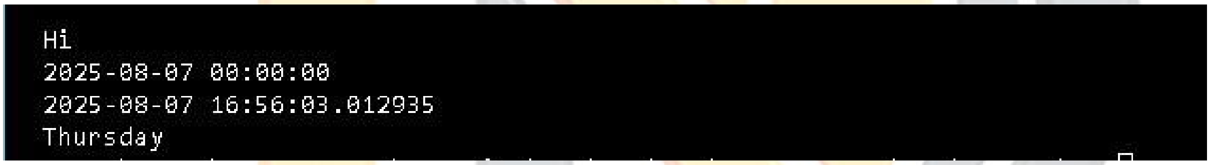
1. Use random & datetime in script.

- In this task I use random and datetime module to get the better understanding of these modules
- By using random the random number will print the specific keyword
- While 2 of datetime functions are used to print the date or displaying current date and day



```python
#using random and date time module
import random
import datetime
num = random.randint(0,4)
if num == 1:
    print('Hello')
elif num == 2:
    print('Hi')
elif num == 3:
    print('Hey')
elif num == 4:
    print('Nice to meet you!')

a = datetime.datetime(2025,8,7)
print(a)
b = datetime.datetime.now()
print(b)
print(b.strftime("%A"))
```

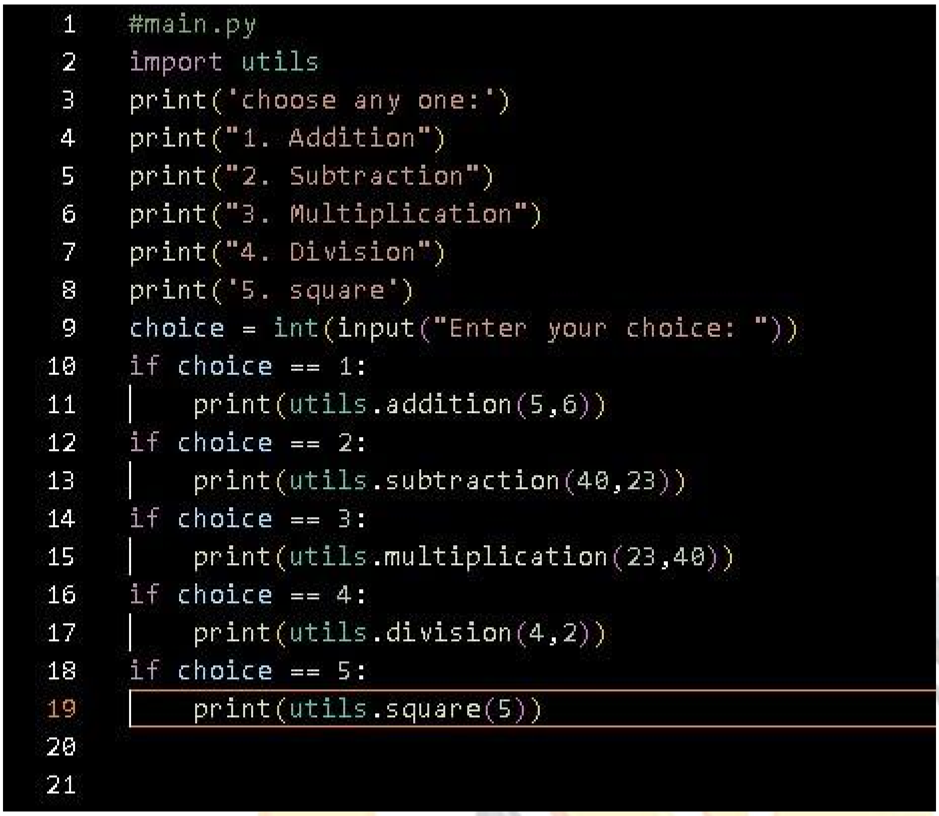
Output:



```
Hi
2025-08-07 00:00:00
2025-08-07 16:56:03.012935
Thursday
```

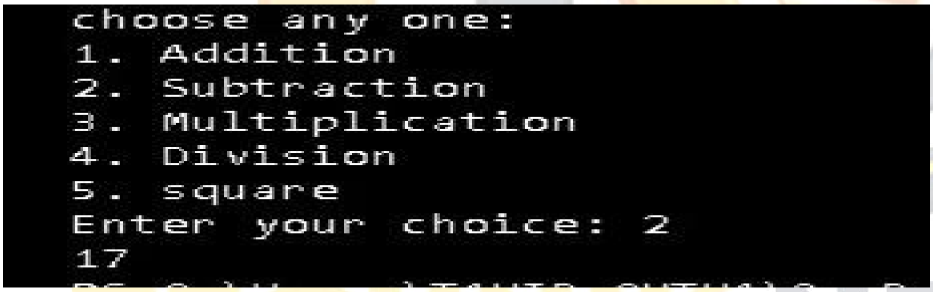
2. Create math_utils module & import.

- Utils file created and import in the other file
- define multiple calculator functions are used to perform operations on the parameter and return the result



```python
#main.py
import utils
print('choose any one:')
print("1. Addition")
print("2. Subtraction")
print("3. Multiplication")
print("4. Division")
print('5. square')
choice = int(input("Enter your choice: "))
if choice == 1:
    print(utils.addition(5,6))
if choice == 2:
    print(utils.subtraction(40,23))
if choice == 3:
    print(utils.multiplication(23,40))
if choice == 4:
    print(utils.division(4,2))
if choice == 5:
    print(utils.square(5))
```

Output:

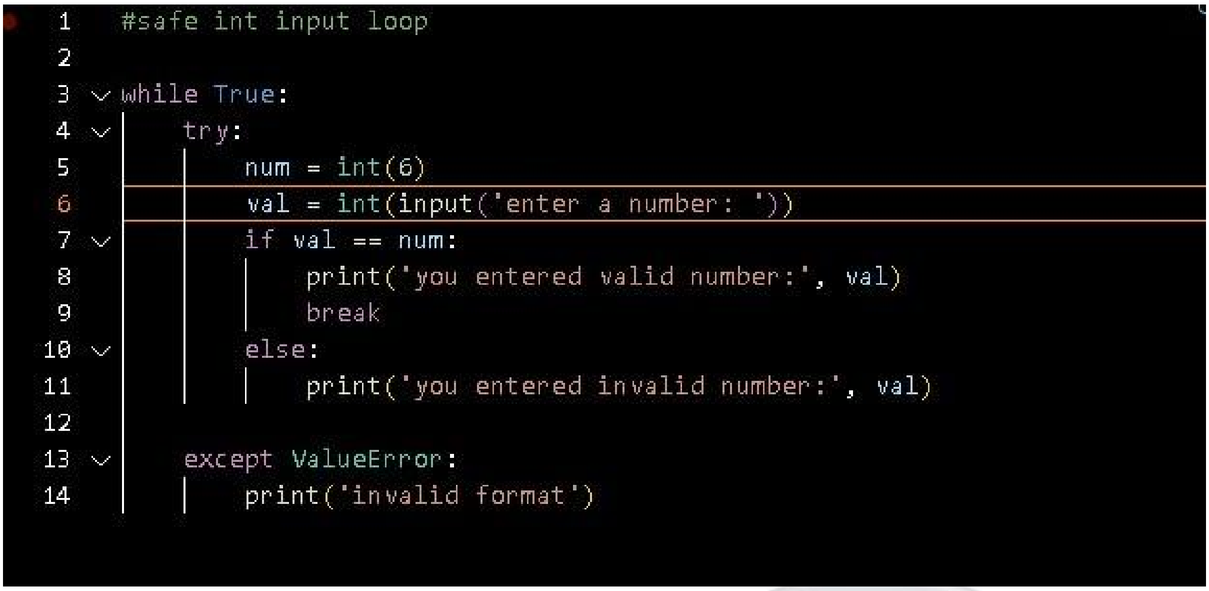


```
choose any one:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. square
Enter your choice: 2
17
```

Tasks –Exceptions

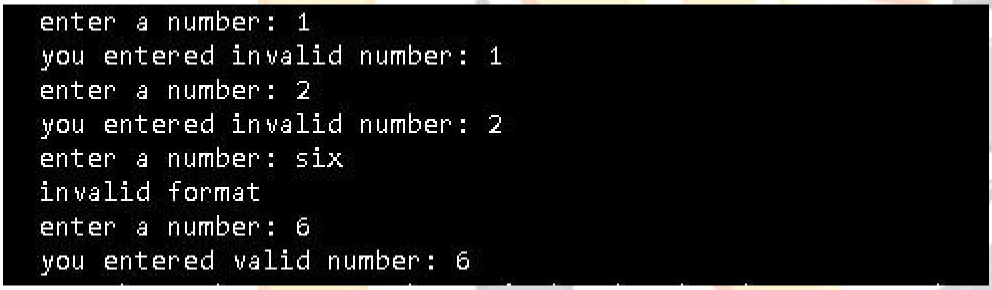
1.Safe int input loop.

- This program take input from the user
- If the user enter the exact saved number
- It shows the result
- Otherwise it runs untill the exact number is entered by the user



```python
#safe int input loop

while True:
    try:
        num = int(6)
        val = int(input('enter a number: '))
        if val == num:
            print('you entered valid number:', val)
            break
        else:
            print('you entered invalid number:', val)

    except ValueError:
        print('invalid format')
```

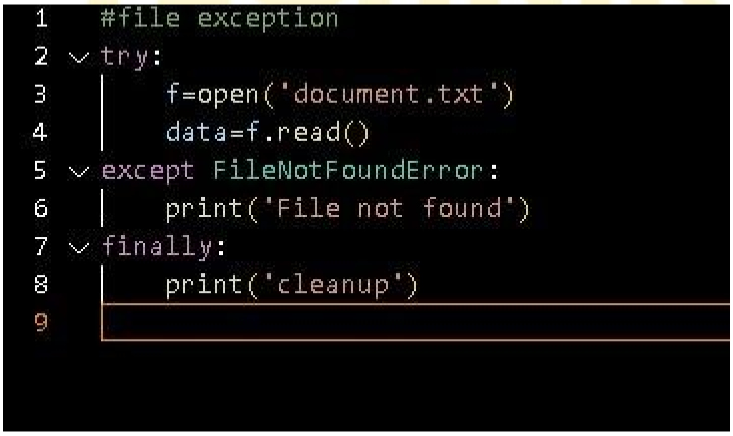
Output:



```
enter a number: 1
you entered invalid number: 1
enter a number: 2
you entered invalid number: 2
enter a number: six
invalid format
enter a number: 6
you entered valid number: 6
```

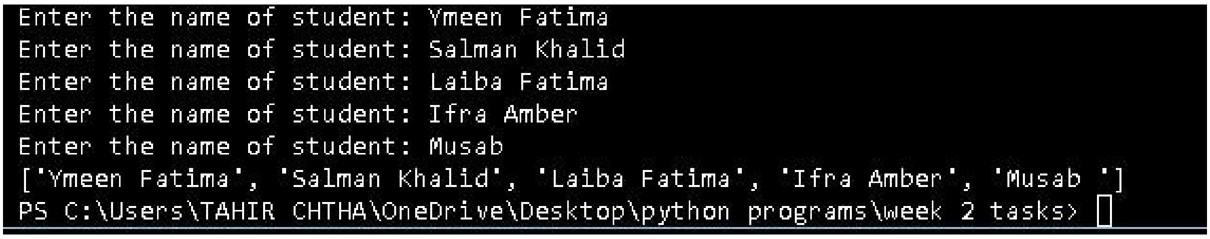
2. File open with error message.

- This shows the file not found error when user tried to open any non existing file



```python
#file exception
try:
    f=open('document.txt')
    data=f.read()
except FileNotFoundError:
    print('File not found')
finally:
    print('cleanup')

```

Output:

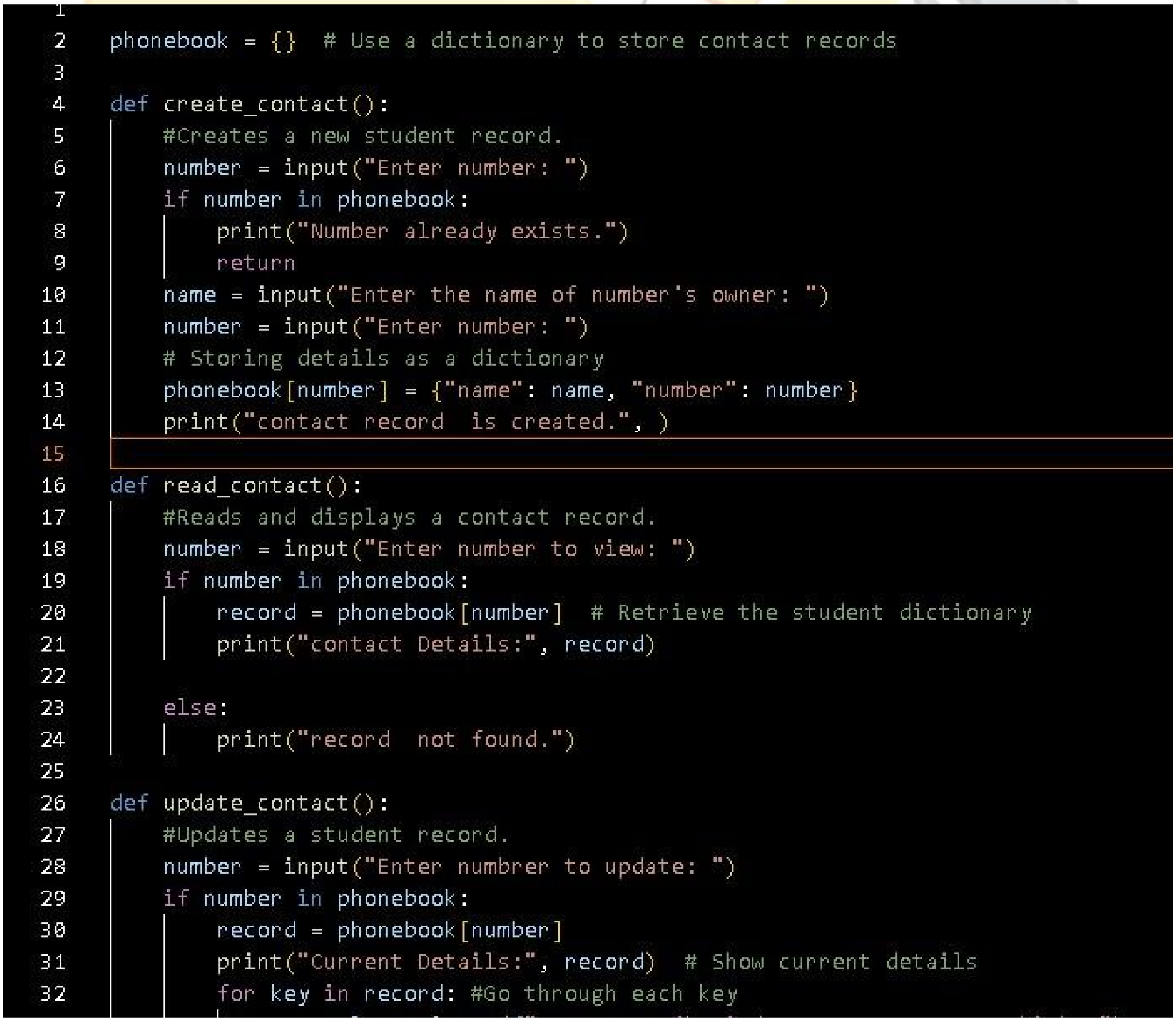


```
Enter the name of student: Ymeen Fatima
Enter the name of student: Salman Khalid
Enter the name of student: Laiba Fatima
Enter the name of student: Ifra Amber
Enter the name of student: Musab
['Ymeen Fatima', 'Salman Khalid', 'Laiba Fatima', 'Ifra Amber', 'Musab ']
PS C:\Users\TAHIR CHTHA\OneDrive\Desktop\python programs\week 2 tasks> 
```

Weekly Challenge (Hard)

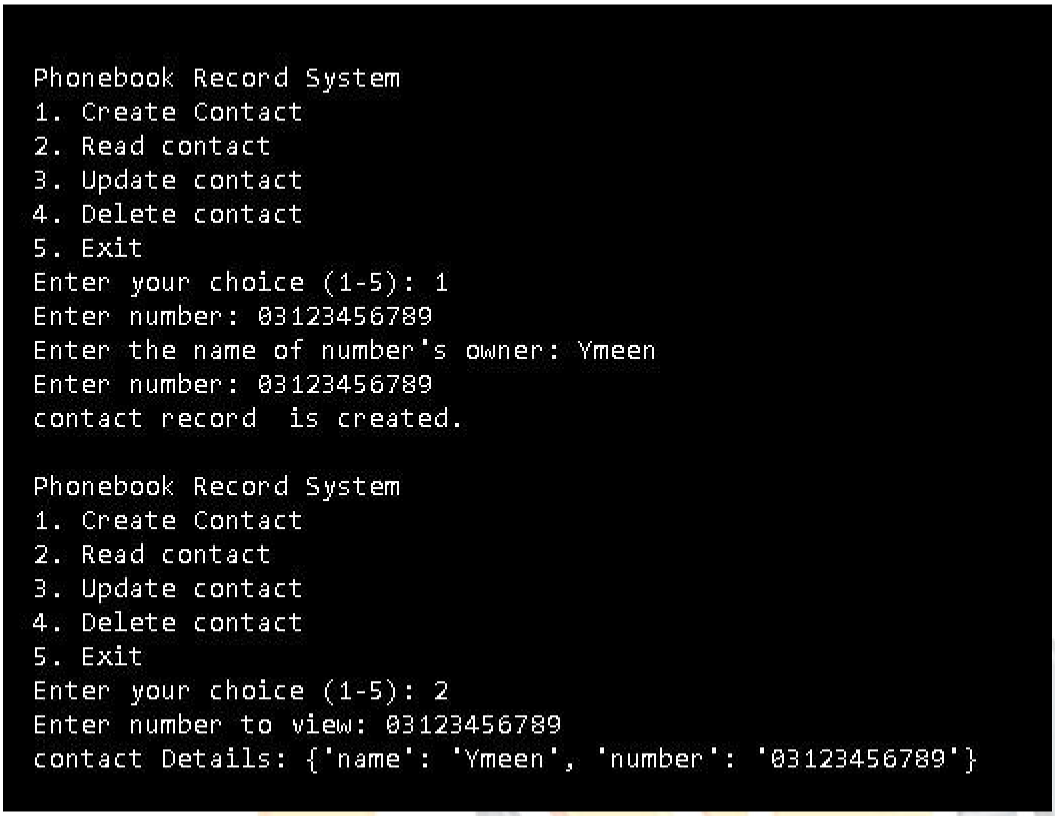
Phonebook App: CRUD contacts dict <-> JSON file storage

- Make a phone book file that stores the information using defined functions
- Number is used as a unique key in the dictionary of phonebook



```

phonebook = {}  # Use a dictionary to store contact records

def create_contact():
    #Creates a new student record.
    number = input("Enter number: ")
    if number in phonebook:
        print("Number already exists.")
        return
    name = input("Enter the name of number's owner: ")
    number = input("Enter number: ")
    # Storing details as a dictionary
    phonebook[number] = {"name": name, "number": number}
    print("contact record  is created.", )

def read_contact():
    #Reads and displays a contact record.
    number = input("Enter number to view: ")
    if number in phonebook:
        record = phonebook[number]  # Retrieve the student dictionary
        print("contact Details:", record)

    else:
        print("record  not found.")

def update_contact():
    #Updates a student record.
    number = input("Enter numbrer to update: ")
    if number in phonebook:
        record = phonebook[number]
        print("Current Details:", record)  # Show current details
        for key in record: #Go through each key
```

OUTPUT:



```
Phonebook Record System
1. Create Contact
2. Read contact
3. Update contact
4. Delete contact
5. Exit
Enter your choice (1-5): 1
Enter number: 03123456789
Enter the name of number's owner: Ymeen
Enter number: 03123456789
contact record  is created.

Phonebook Record System
1. Create Contact
2. Read contact
3. Update contact
4. Delete contact
5. Exit
Enter your choice (1-5): 2
Enter number to view: 03123456789
contact Details: {'name': 'Ymeen', 'number': '03123456789'}
```

Learning and chalanges:

I have learn many concepts in this week tasks while there are some tasks that I felt hard to implement because I’m at beginner level and didn't know their logic although these tasks help me understand python in depth.

The most difficult part is to join python file with json and I wasn’t able to complete this task because of my limited knowledge.hope the instructor will address this difficulty.