

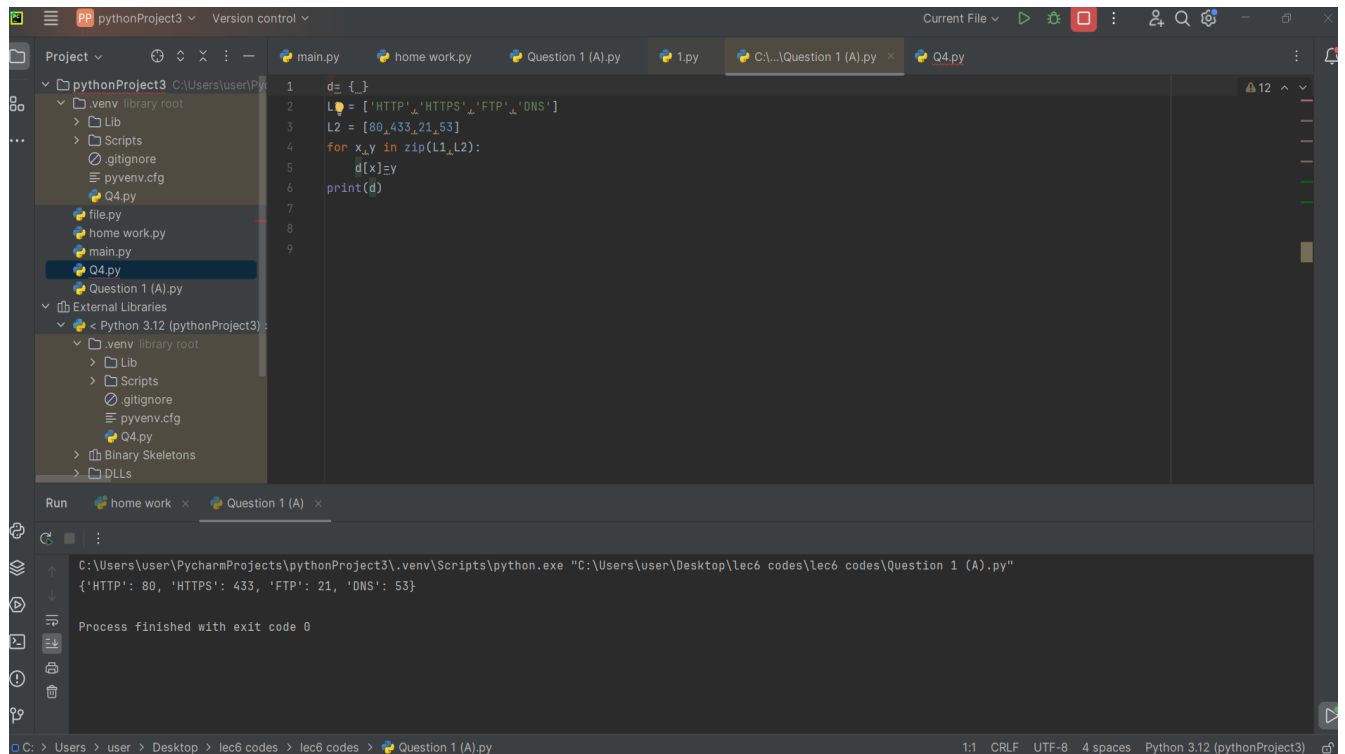
Name : Maya Mahmoud

Number :2828

Submitted to GitHub :

Question 1:

A–If you have two lists, L1=['HTTP','HTTPS','FTP','DNS']
L2=[80,443,20,53], convert it to generate this dictionary
d={'HTTP':80,'HTTPS':443,'FTP':21,'DNS':53 }



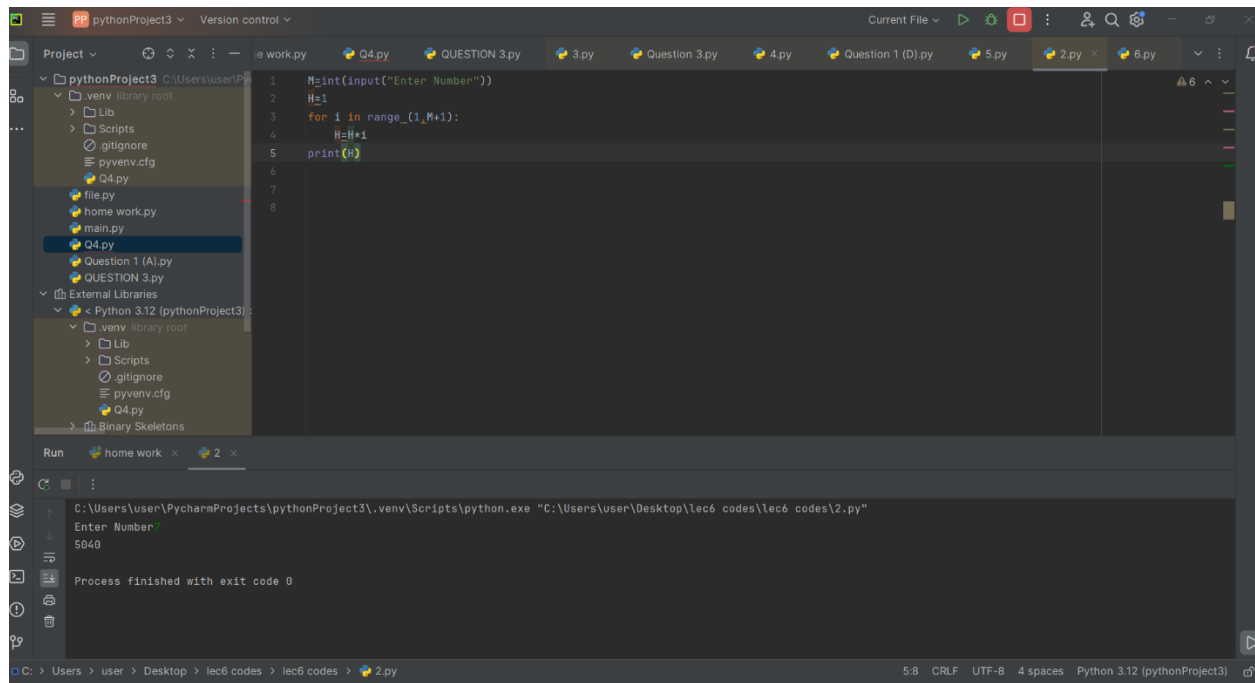
The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script named 'Question 1 (A).py' with the following code:

```
1 d= {}
2 L1 = ['HTTP','HTTPS','FTP','DNS']
3 L2 = [80,443,21,53]
4 for x,y in zip(L1,L2):
5     d[x]=y
6 print(d)
```

The left sidebar shows the project structure for 'pythonProject3', including files like 'main.py', 'home work.py', 'Question 1 (A).py', and 'Q4.py'. The bottom panel shows the 'Run' output, indicating that the script was executed successfully and produced the expected dictionary output:

```
C:\Users\user\PycharmProjects\pythonProject3\.venv\Scripts\python.exe "C:\Users\user\Desktop\lec6 codes\lec6 codes\Question 1 (A).py"
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
Process finished with exit code 0
```

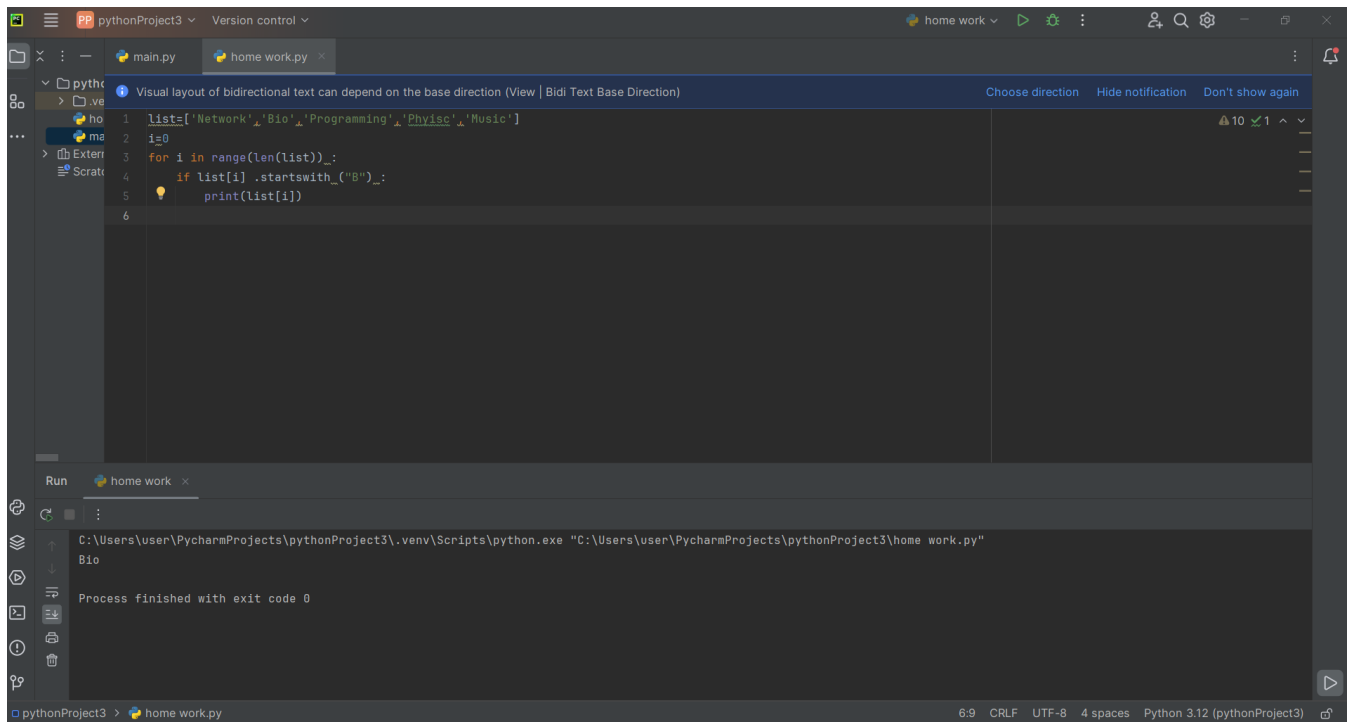
B– Write a Python program that calculates the factorial of a given number entered by user.



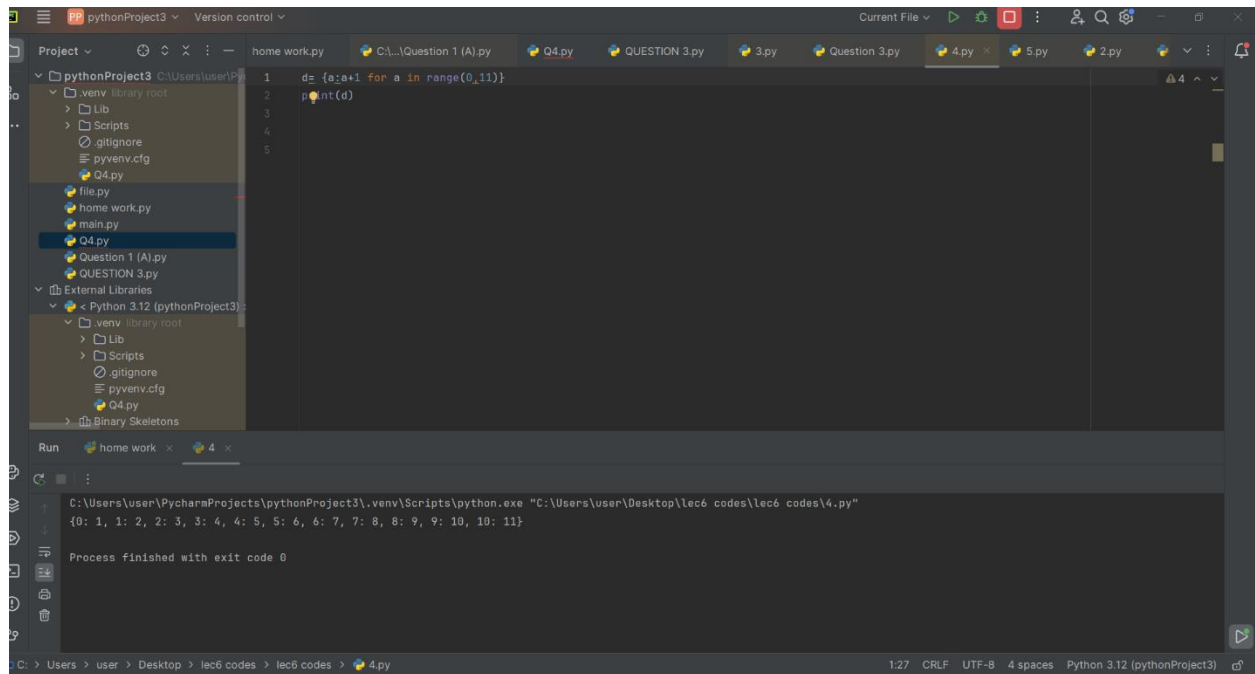
C- L=['Network' , 'Bio' , 'Programming' , 'Physics' , 'Music']

In this exercise, you will implement a Python program that reads the items of the previous list and identifies the items that starts with 'B' letter, then print it on screen.

Tips: using loop, 'Len ()' , starts with() method



D: Using Dictionary comprehension, Generate this dictionary
`d={0:1,1:2,2:3,3:4,4:5,5:6,6:7,7:8,8:9,9:10,10:11}`



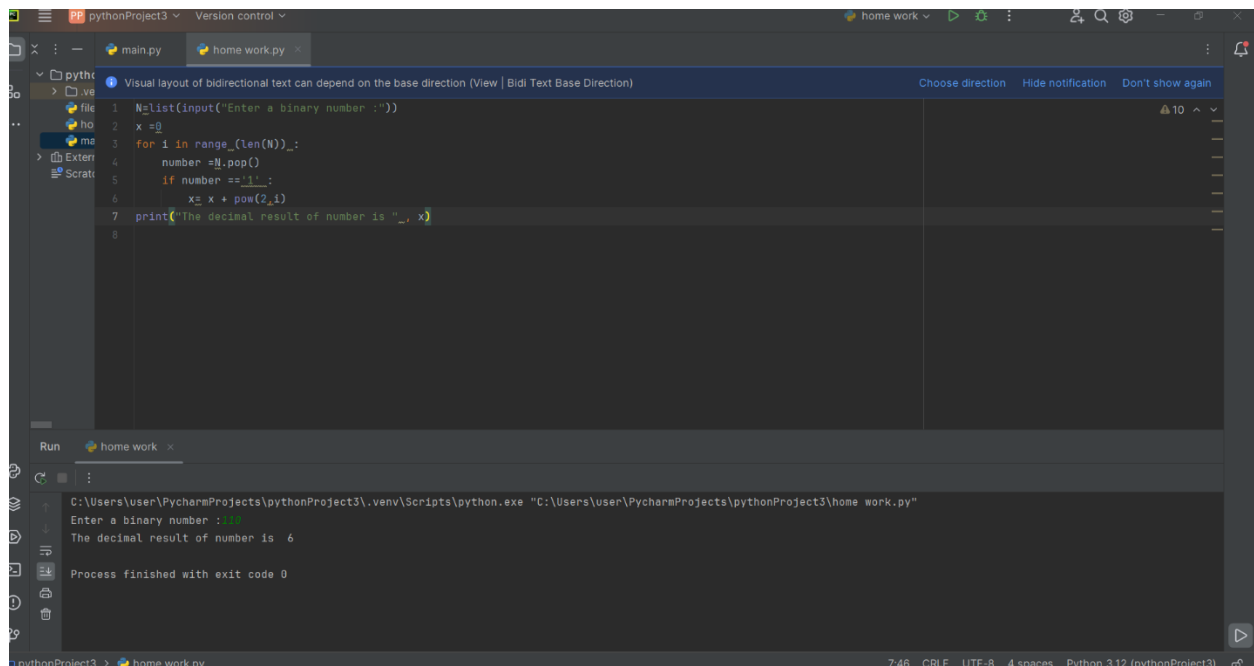
Question 2:

Convert from Binary to Decimal.

Write a Python program that converts a Binary number into its equivalent Decimal number.

The program should start reading the binary number from the user. Then the decimal equivalent number must be calculated. Finally, the program must display the equivalent decimal number on the screen.

Tips: solve input errors.



```
1 N=list(input("Enter a binary number :"))
2 x=0
3 for i in range(len(N)):
4     number=N.pop()
5     if number=="1":
6         x=x+pow(2,i)
7 print("The decimal result of number is :",x)
```

Run home work

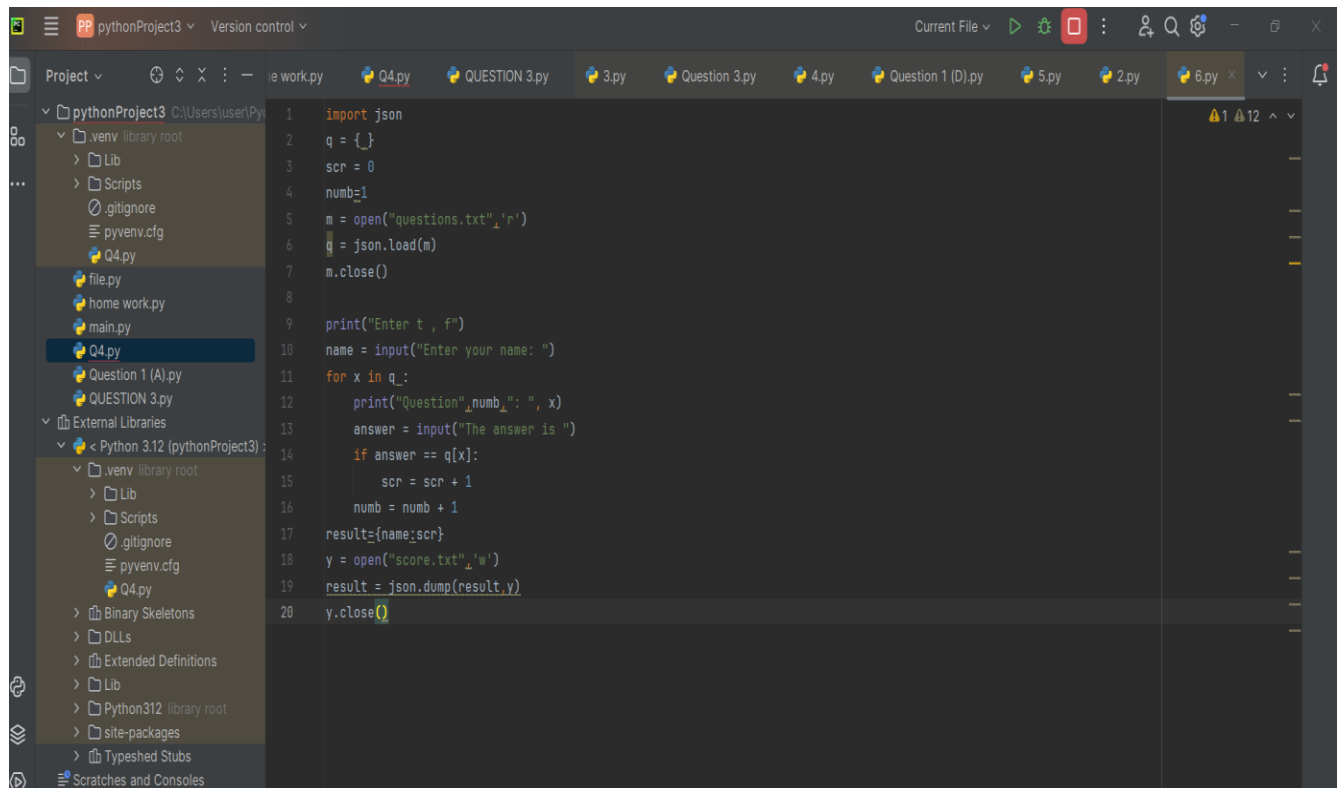
```
C:\Users\User\PycharmProjects\pythonProject3\.venv\Scripts\python.exe "C:\Users\User\PycharmProjects\pythonProject3\home work.py"
Enter a binary number : 110
The decimal result of number is 6

Process finished with exit code 0
```

Question 3:

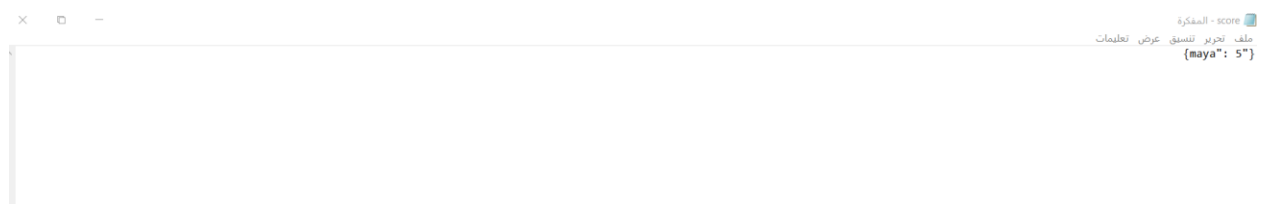
Working with Files" Quiz Program".

Type python quiz program that takes a text or json or csv file as input for (20 (Questions, Answers)). It asks the questions and finally computes and prints user results and store user name and result in separate file csv or json file.



The screenshot shows a code editor with a Python file named Q4.py. The code is a quiz program that reads questions from a JSON file, asks them to the user, and saves the results to a JSON file. The code is as follows:

```
1 import json
2 q = {}
3 scr = 0
4 numb=1
5 m = open("questions.txt", 'r')
6 q = json.load(m)
7 m.close()
8
9 print("Enter t , f")
10 name = input("Enter your name: ")
11 for x in q_:
12     print("Question", numb,": ", x)
13     answer = input("The answer is ")
14     if answer == q[x]:
15         scr = scr + 1
16         numb = numb + 1
17 result={name:scr}
18 y = open("score.txt", 'w')
19 result = json.dump(result, y)
20 y.close()
```



Question 4: Object-Oriented Programming - Bank Class

Define a class Bank Account with the following attributes and methods:

Attributes: account_number (string), account_holder (string), balance (float, initialized to 0.0)

Methods: deposit(amount), withdraw(amount), get_balance()

- Create an instance of Bank Account, - Perform a deposit of \$1000, - Perform a withdrawal of \$500.

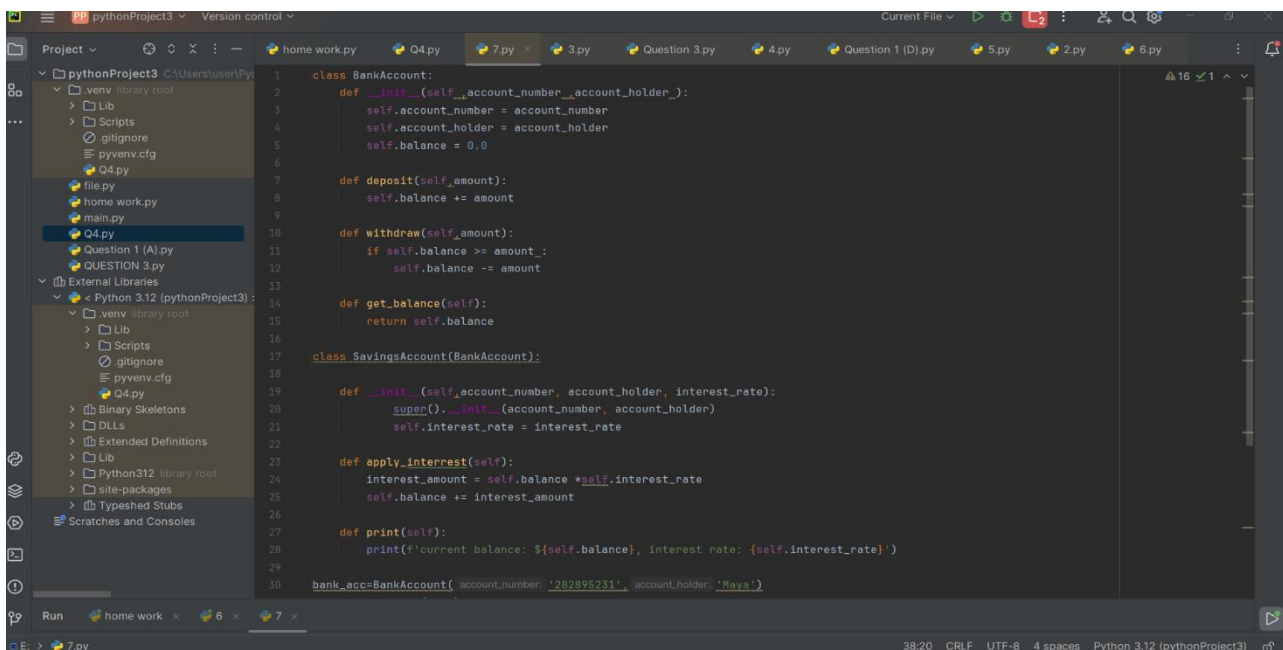
- Print the current balance after each operation.

- Define a subclass Savings Account that inherits from Bank Account and adds interest_rate Attribute and

apply_interest() method that Applies interest to the balance based on the interest rate.

And Override print() method to print the current balance and rate.

- Create an instance of Savings Account, and call apply_interest() and print() functions.



```
1 class BankAccount:
2     def __init__(self, account_number, account_holder):
3         self.account_number = account_number
4         self.account_holder = account_holder
5         self.balance = 0.0
6
7     def deposit(self, amount):
8         self.balance += amount
9
10    def withdraw(self, amount):
11        if self.balance >= amount:
12            self.balance -= amount
13
14    def get_balance(self):
15        return self.balance
16
17    class SavingsAccount(BankAccount):
18
19        def __init__(self, account_number, account_holder, interest_rate):
20            super().__init__(account_number, account_holder)
21            self.interest_rate = interest_rate
22
23        def apply_interest(self):
24            interest_amount = self.balance * self.interest_rate
25            self.balance += interest_amount
26
27        def print(self):
28            print(f'current balance: ${self.balance}, interest rate: {self.interest_rate}')
29
30    bank_acc = BankAccount(account_number='282895231', account_holder='Maya')
```

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, running, and debugging. The 'Project' view on the left shows the project structure for 'pythonProject3', including a virtual environment and various Python files. The main editor displays the code for '7.py', which defines a 'SavingsAccount' class and performs several operations. The 'Run' view at the bottom shows the execution output, including the path to the Python interpreter and the results of the program's execution.

```
17 class SavingsAccount(BankAccount):
18     def __init__(self, account_number, account_holder, interest_rate):
19         super().__init__(account_number, account_holder)
20         self.interest_rate = interest_rate
21
22
23     def apply_interest(self):
24         interest_amount = self.balance * self.interest_rate
25         self.balance += interest_amount
26
27     def print(self):
28         print(f'current balance: ${self.balance}, interest rate: {self.interest_rate}')
29
30 bank_acc = BankAccount('282895231', 'Nave')
31 bank_acc.deposit(1000)
32 print(f'balance after deposit: $ {bank_acc.get_balance()}')
33 bank_acc.withdraw(500)
34 print(f'Balance after withdraw: $ {bank_acc.get_balance()}')
35 savings_acc = SavingsAccount('228866888', 'Mera', interest_rate=0.07)
36 savings_acc.deposit(2000)
37 savings_acc.apply_interest()
38 savings_acc.print()
```

Run
C:\Users\User\PycharmProjects\pythonProject3\.venv\Scripts\python.exe E:\7.py
balance after deposit: \$ 1000.0
Balance after withdraw: \$ 500.0
current balance: \$2140.0, interest rate: 0.07
Process finished with exit code 0