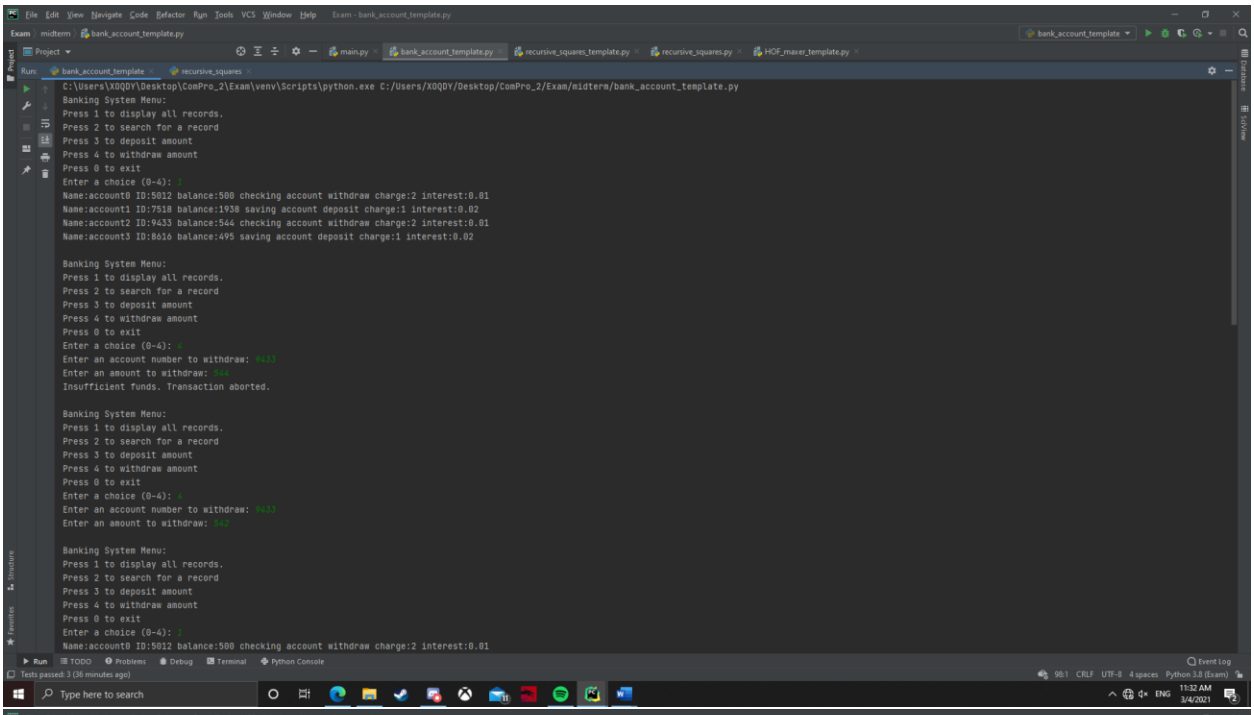
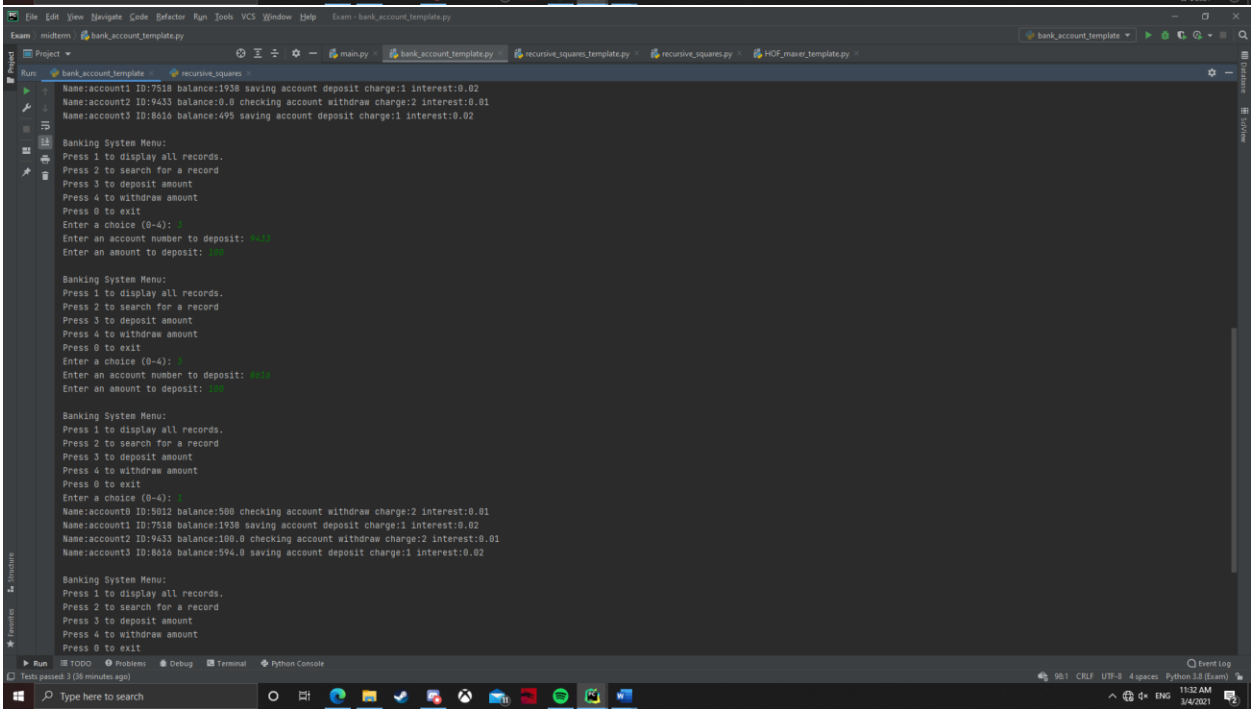
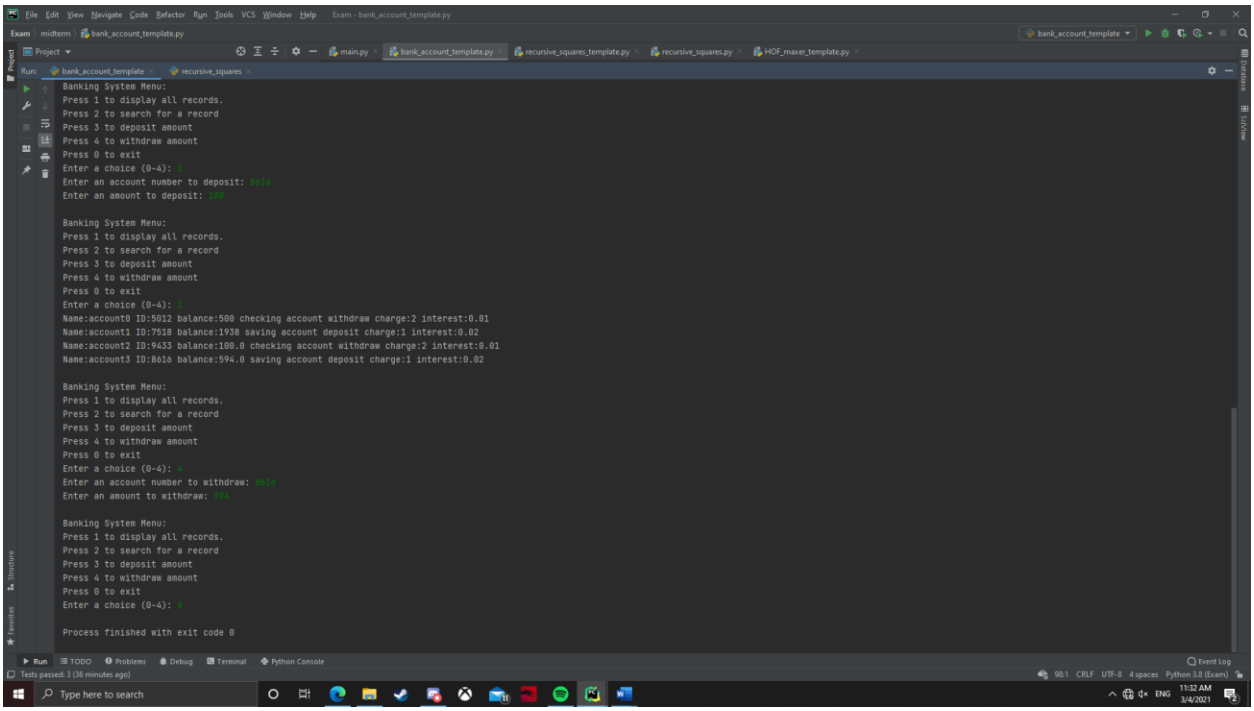
bank_account_template

For bank_account_template for class CheckingAccount it simple I just write __init__ by using super().__init__ and modify a display function and withdraw function by change a little by you can see that from my screenshot. For class SavingsAccount I just do like CheckingAccount different for this class is change depoit function not withdraw

```
class CheckingAccount(BankAccount):
    def __init__(self, acc_id, name, balance):
        super().__init__(acc_id, name, balance)

    def withdraw(self, amount):
        amount += 2
        if self.balance < amount:
            print('Insufficient funds. Transaction aborted.')
            raise Exception()
        else:
            self.balance -= amount

    def display(self):
        print("Name:" + self.account_name, "ID:" + self.account_id, "balance:" + str(self.balance),
              "checking account withdraw charge:2 interest:0.01", end='')
        print()


class SavingsAccount(BankAccount):
    def __init__(self, acc_id, name, balance):
        super().__init__(acc_id, name, balance)

    def deposit(self, amount):
        self.balance += amount - 1

    def display(self):
        print("Name:" + self.account_name, "ID:" + self.account_id, "balance:" + str(self.balance),
              "saving account deposit charge:1 interest:0.02", end='')
        print()
```


C:\Users\XOQDY\Desktop\ComPro_2\Exam\venv\Scripts\python.exe C:/Users/XOQDY/Desktop/ComPro_2/Exam/midterm/bank_account_template.py
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 1
Name:account0 ID:5012 balance:500 checking account withdraw charge:2 interest:0.01
Name:account1 ID:7518 balance:1938 saving account deposit charge:1 interest:0.02
Name:account2 ID:9433 balance:544 checking account withdraw charge:2 interest:0.01
Name:account3 ID:8616 balance:495 saving account deposit charge:1 interest:0.02
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 4
Enter an account number to withdraw: 9433
Enter an amount to withdraw: 5000
Insufficient funds. Transaction aborted.
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 4
Enter an account number to withdraw: 9433
Enter an amount to withdraw: 544
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 1
Name:account0 ID:5012 balance:500 checking account withdraw charge:2 interest:0.01

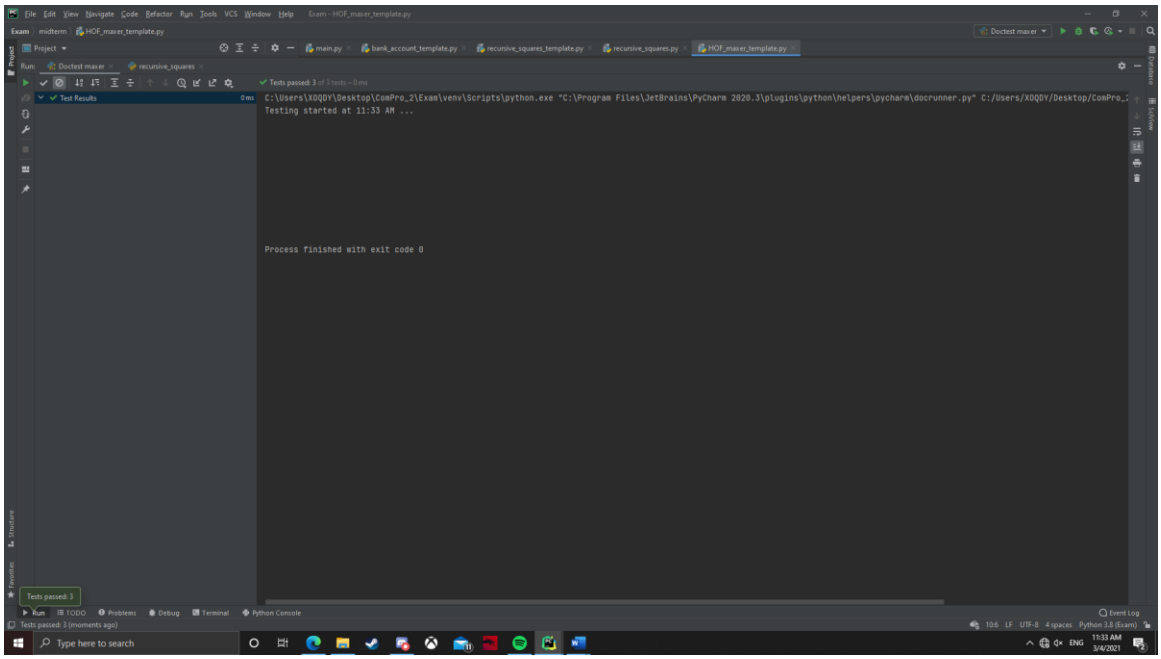

Name:account1 ID:7518 balance:1938 saving account deposit charge:1 interest:0.02
Name:account2 ID:9433 balance:0.0 checking account withdraw charge:2 interest:0.01
Name:account3 ID:8616 balance:495 saving account deposit charge:1 interest:0.02
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 3
Enter an account number to deposit: 9433
Enter an amount to deposit: 100
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 3
Enter an account number to deposit: 8616
Enter an amount to deposit: 100
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 1
Name:account0 ID:5012 balance:500 checking account withdraw charge:2 interest:0.01
Name:account1 ID:7518 balance:1938 saving account deposit charge:1 interest:0.02
Name:account2 ID:9433 balance:100.0 checking account withdraw charge:2 interest:0.01
Name:account3 ID:8616 balance:594.0 saving account deposit charge:1 interest:0.02
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit


Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 3
Enter an account number to deposit: 8410
Enter an amount to deposit: 100
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 1
Name:account0 ID:5012 balance:500 checking account withdraw charge:2 interest:0.01
Name:account1 ID:7518 balance:1938 saving account deposit charge:1 interest:0.02
Name:account2 ID:9433 balance:100.0 checking account withdraw charge:2 interest:0.01
Name:account3 ID:8610 balance:594.0 saving account deposit charge:1 interest:0.02
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 4
Enter an account number to withdraw: 8610
Enter an amount to withdraw: 594
Banking System Menu:
Press 1 to display all records.
Press 2 to search for a record
Press 3 to deposit amount
Press 4 to withdraw amount
Press 0 to exit
Enter a choice (0-4): 0
Process finished with exit code 0

HOF_maxer_template

For fire function I use it to print out the largest smoke(y) and haze function first I check if the new value is greater or not if it I will return mixer(smoke)(z) if not just return mixer(smoke)(y) not z

```
def maxer(smoke):
    """Return a repeatable function fire(y) that prints the largest smoke(y) so far.
    >>> g = maxer(square)
    >>> h = g(2)(1)(3)(2)(-4) # print the largest square(y) so far
    4
    4
    9
    9
    16
    >>> h = maxer(abs)(2)(1)(3)(2)(-4) # print the largest abs(y) so far
    2
    2
    3
    3
    4
    """
    def fire(y):
        # fill one line of code here
        print(smoke(y))
        def haze(z):
            # fill an if condition and a line of code in the if block here
            if smoke(z) > smoke(y):
                return maxer(smoke)(z)
            # fill a return line here
            return maxer(smoke)(y)
        return haze
    return fire
```



recursive_squares

It's simple I just call recursive_draw_square again and each time I call it I will -1 level and /2 size then for x, y first I decide to draw top left that for x – size/2, y + size/2 After that followed by top right(x + size/2, y + size/2), bottom left(x – size/2, y – size/2), bottom right(x + size/2, y – size/2) respectively

```
def recurse_draw_square(level, size, x, y):
    if level == 0:
        return

    turtle.penup()
    turtle.goto(x, y)
    draw_square(size)
    # you fill in the rest of the code
    recurse_draw_square(level - 1, size / 2, x - size / 2, y + size / 2)
    recurse_draw_square(level - 1, size / 2, x + size / 2, y + size / 2)
    recurse_draw_square(level - 1, size / 2, x - size / 2, y - size / 2)
    recurse_draw_square(level - 1, size / 2, x + size / 2, y - size / 2)
```