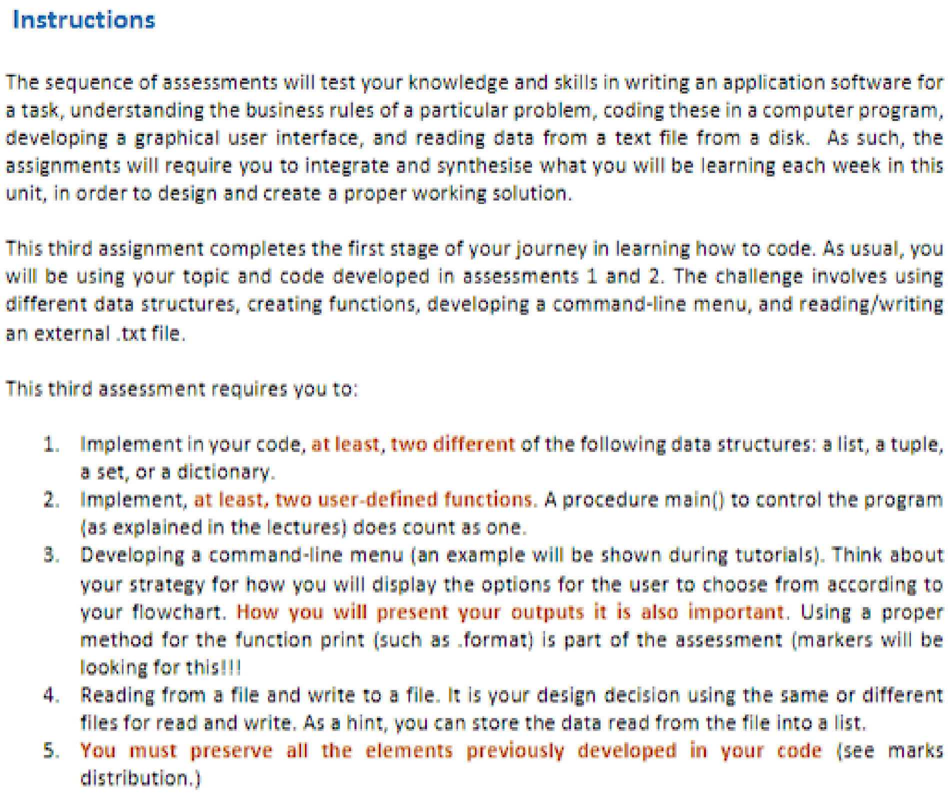
UNIVERSITY OF CANBERRA
INTRODUCTION TO INFORMATION TECHNOLOGY (4478)

Alex Greenfield u3242795 Due date: 3/4/2022 Julio Romero



Instructions

The sequence of assessments will test your knowledge and skills in writing an application software for a task, understanding the business rules of a particular problem, coding these in a computer program, developing a graphical user interface, and reading data from a text file from a disk. As such, the assignments will require you to integrate and synthesise what you will be learning each week in this unit, in order to design and create a proper working solution.

This third assignment completes the first stage of your journey in learning how to code. As usual, you will be using your topic and code developed in assessments 1 and 2. The challenge involves using different data structures, creating functions, developing a command-line menu, and reading/writing an external .txt file.

This third assessment requires you to:

1. Implement in your code, at least, two different of the following data structures: a list, a tuple, a set, or a dictionary.
2. Implement, at least, two user-defined functions. A procedure main() to control the program (as explained in the lectures) does count as one.
3. Developing a command-line menu (an example will be shown during tutorials). Think about your strategy for how you will display the options for the user to choose from according to your flowchart. How you will present your outputs it is also important. Using a proper method for the function print (such as .format) is part of the assessment (markers will be looking for this!!!
4. Reading from a file and write to a file. It is your design decision using the same or different files for read and write. As a hint, you can store the data read from the file into a list.
5. You must preserve all the elements previously developed in your code (see marks distribution.)

I) Background: **Five [5]marks for an improved background about the problem proposed or for the suitable background of your choosing topic.**

Python can be very useful to create calculators, especially in finance where you can create programs to automate particular problems, instead of manually calculating, in our case 3 banks and finding their APY and balance. We can create a program/module so that the user does not have to manually calculate these problems and shorten the process for the user by using the program, after all inputs are entered into the module, the answer will be output.

The Module in Python language should compare interest rates offered by three different banks. Determine the most favorable interest rate and the balance in that account after one year. Create a calculator in python to do this for the user, printing the solution as an output in various ways. Such as a shell-based menu to choose from, and writing the output to a text file/files. The list should be read from a text file. In our case, this is a list of the banks. This is displayed above the menu as a list, of the available options to choose from.

II) Business Rules (based on the 5-steps solving - problems process) [Total marks:25]

1. Identify your inputs. [MARKS: 5 marks for a clear and specific description of updated inputs]

Input 1 = **r** (interest in decimal form (2dp)) = *0.04, 0.05, 0.03*
Input 2 = **m** (compound interval in a year, in months) = *8, 4, 12*
Input 3 = **P** *(deposit ($)) = $2000*

2. Identify the goal or objective (related to the output) [MARKS: 5 marks for a clear and specific description of updated outputs]

We need to assess the problem, and create a solution in the form of a calculator that outputs our balance and APY percentage.

We need to use the APY formula to compare the banks and find the highest APY percentage among the 3 banks. Outputting the results. Different compounding periods cannot be compared directly that is why we must use APY. The following formula is to be implemented into the calculator below.

$$APY = \left(1 + \frac{r}{m}\right)^{m} - 1$$

$$P \cdot \left(1 + \frac{r}{m}\right)^{m}$$

Details: (*Bank - 1*, **P** = $2000, **r** = 0.04%, **m** = 8) (*Bank* - *2*, **P** = $2000, **r** = 0.05%, **m** = 4) *(Bank - 3* **P** = $2000, **r** = 0.03%, **m** = 12)

Compare banks, find the most favorable bank and it's interest rate using the APY formula. Output the bank, balance and APY % in that account after one year.

3. Create a list of tasks to achieve your objective [MARKS: 5 marks for an updated sequence of tasks]

1. Create a flow chart, develop pseudocode from the flowchart, identify inputs and outputs, variables, constants decisions, loops, booleans, strings, integers, etc. (see rubric for more details)

2. Code the solution into Python IDLE, using the pseudocode as a reference. Use your flowchart as a reference to code your solution into python more easily. This process helps in the designing and planning of your code when it comes time to write it. It is important to eliminate easy mistakes and problems.

   3. Code inputs in (**r(1,2,3), m(1,2,3)**), floats for interest, interest and principal. Code in elements such as lists, booleans, decision, while and for loops, etc. Added a shell-based menu and at least one write/read text file to write the outputs into.

   4. Make sure code follows the Python Style Guide for easy writing/reading and interpretation of the code. This makes sure the code is neat and tidy. This will help, and sort your code into sections, overall this helps the creator of the code and the reader.

4. Develop a hand-written solution **[MARKS: 5 marks for an updated written solution]**

Using the inputs interest, principle, and compound interval. Solve the best bank balance and APY % between the 3 banks.

P = Principal
r = Interest
M = Compound Interval

Commonwealth Bank: $2000, 4%, 8
Beyond Bank: $2000, 5%, 4
St George Bank: $2000, 3%, 12

Formula for Balance ($):

$$P \cdot \left(1+\frac{r}{m}\right)^{m}$$

Commonwealth Bank = = $2081.41
Beyond Bank =) = $2101.89
St George Bank = = $2060.83

Formula for APY (%):

$$APY = \left(1+\frac{r}{m}\right)^{m} - 1$$

Commonwealth Bank (1) = 4%
Beyond Bank (2) = 5%
St George Bank (3) = 3%

Compare banks with the APY formula, not with the balance. Output the best bank option, balance, and APY.

Solution: Bank 2 had the best APY, 5% interest, compounded quarterly. $2000 dollars principal, $2101.89 after one year.

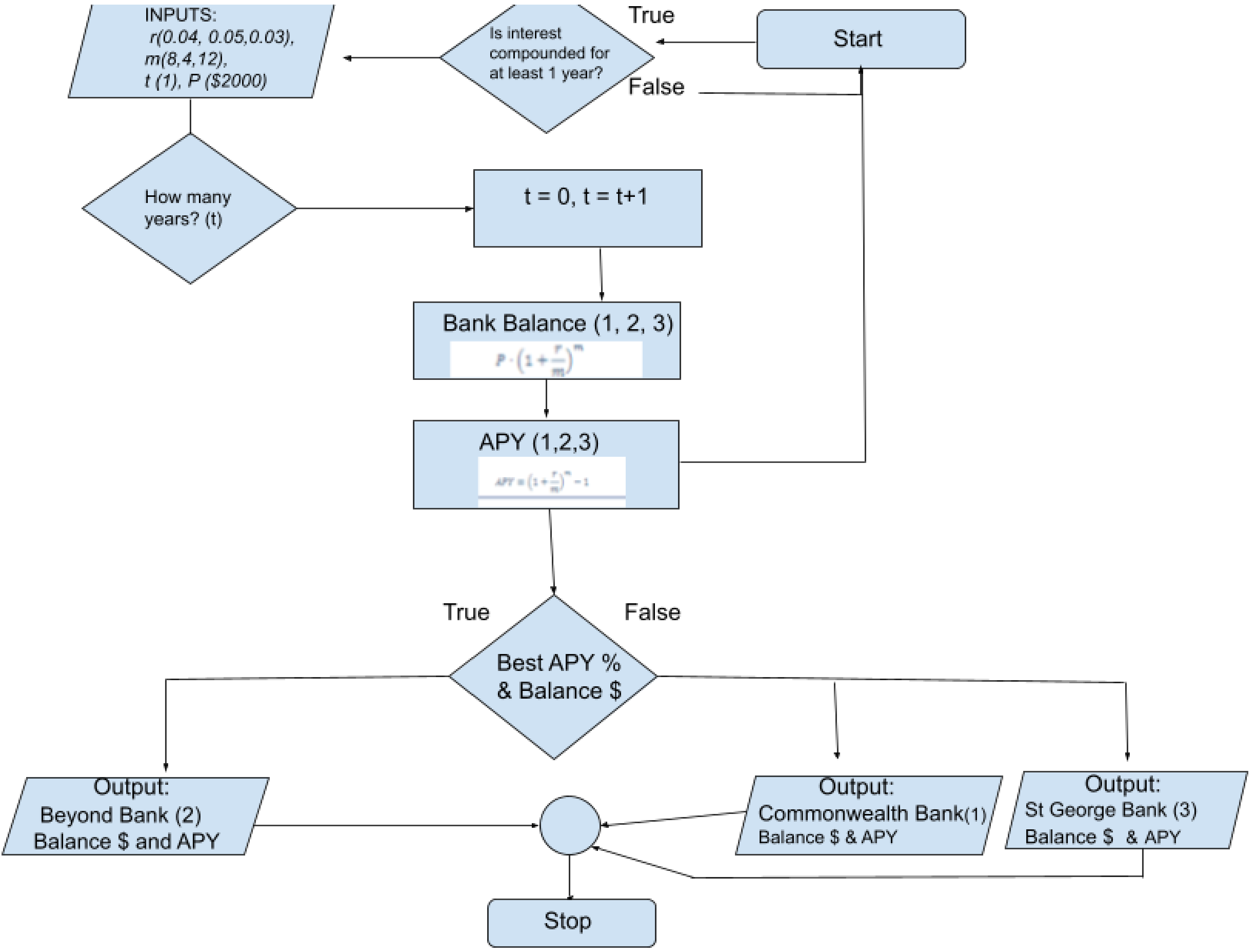
5. Assess (test) your solution **[MARKS: 5 marks for properly assessing your solution]**

1. Handwritten solution tested with a scientific calculator, handwritten results checked.
2. Use a Balance & APY calculator to test the results, tested and confirmed.
3. Inputs and output assessed and tested, correct.
4. Test our solution, in our own calculator and compare results. Results match the handwritten solution.
5. Tests conducted. Results match. Python calculator outputs correctly (TUI).
6. Shell-based menu outputs a correct answer, along with the best bank & banks in their own individual text files (TUI).

III) The process to Start Coding your Solution **[Total marks: 70]**

6. Produce a flowchart to describe the flow of instructions

[MARKS: 8 marks for a properly designed flowchart]


Start
True
Is interest compounded for at least 1 year?
False
INPUTS:
r(0.04, 0.05,0.03),
m(8,4,12),
t (1), P ($2000)
How many years? (t)
t = 0, t = t+1
Bank Balance (1, 2, 3)
$P \cdot \left(1+\frac{r}{m}\right)^m$
APY (1,2,3)
$APY = \left(1+\frac{r}{m}\right)^m - 1$
True
False
Best APY % & Balance $
Output: Beyond Bank (2) Balance $ and APY
Output: Commonwealth Bank(1) Balance $ & APY
Output: St George Bank (3) Balance $ & APY
Stop

7. Derive pseudocode following step 6 **[MARKS: 7 marks for a properly designed pseudocode]**

Pseudocode:

```
## Date last changed:8/04/2022
## Author: Alex Greenfield
## Date created: 15/03/2022
## This program is an APY calculator, comparing 3 different banks.
## Inputs: r, m, P, Output: APY % & Balance $
## Balance ($) & APY (%) Calculator: Commonwealth Bank (1), Beyond Bank (2), St George(3)
## Read Input: banklist.txt, Write Outputs: bestbank.txt, banks.txt


SET COMMONWEALTH_BANK TO "Commonwealth Bank"
SET BEYOND_BANK TO "Beyond Bank"
SET STGEORGE_BANK TO "St George Bank"


#Bank1


SET Question1 TO INPUT("Is interest being compounded FOR atleast 1 year?(True/False):")
IF Question1 EQUALS "True":
    SET boolean TO True
ELSEIF Question1 EQUALS "False":
    SET boolean TO False
ELSE:
    SET boolean TO "Error"
    OUTPUT("Please INPUT (True/False)?")
```

```
IF boolean EQUALS True:
    T=float(INPUT("How many years will the interest be compounded?-"))
    P=(float(INPUT("Insert Principal $-")))


    r1=(float(INPUT("Insert Commbank interest rate=")))
    m1=(float(INPUT("Insert Commbank Compound Interval=")))


SET #B1 TO p*(1+r1/m1)**m1


SET #APY1 TO (1+r1/m1)**m1-1



SET t TO 0



WHILE t<= T:
        SET B1 TO P*(1+r1/m1)**m1
        SET APY1 TO (1+r1/m1)**m1-1
        SET t TO t+ 1
        OUTPUT("year =", t)


        OUTPUT("$", B1)
        OUTPUT("%", 100*APY1)
        break



#Bank2
```

```
SET Question2 TO INPUT("Is interest being compounded FOR atleast 1 year?(True/False):")
IF  Question2 EQUALS "True":
     SET boolean TO True
ELSEIF Question2 EQUALS "False":
     SET boolean TO False
ELSE:
    SET boolean TO "Error"
    OUTPUT("Please INPUT (True/False)?")


IF boolean EQUALS True:
    T =float(INPUT("How many years will the interest be compounded?-"))
    P =(float(INPUT("Insert Principal $-")))


    r2=(float(INPUT("Insert Beyond Bank interest rate=")))
    m2=(float(INPUT("Insert Beyond Bank Compound Interval=")))



SET #B2 TO p*(1+r2/m2)**m2


SET #APY2 TO (1+r2/m2)**m2-1



    SET t TO 0


WHILE t<= T:
        SET B2 TO P*(1+r2/m2)**m2
        SET APY2 TO (1+r2/m2)**m2-1

        SET t TO t+ 1

        OUTPUT("year =", t)


        OUTPUT("$", B2)

        OUTPUT("%", 100*APY2)

        break


SET str TO 'Beyond Bank'


#Bank3


SET Question3 TO INPUT("Is interest being compounded FOR atleast 1 year?(True/False):")

IF  Question3 EQUALS "True":

     SET boolean TO True

ELSEIF Question3 EQUALS "False":

    SET boolean TO False

ELSE:

    SET boolean TO "Error"

    OUTPUT("Please INPUT (True/False)?")


IF boolean EQUALS True:

    T =float(INPUT("How many years will the interest be compounded?-"))

    P =(float(INPUT("Insert Principal $-")))


    r3=(float(INPUT("Insert St George interest rate=")))
```

```
   m3=(float(INPUT("Insert St George Bank Compound Interval=")))


SET #B3 TO p*(1+r3/m3)**m3


SET #APY3 TO (1+r3/m3)**m3-1



   SET t TO 0


WHILE t<= T:
      SET B3 TO P*(1+r3/m3)**m3
      SET APY3 TO (1+r3/m3)**m3-1
      SET t TO t+ 1
      OUTPUT("year =", t)


      OUTPUT("$", B3)
      OUTPUT("%", 100*APY3)
      break


#APY Output


SET list1 TO [APY1,APY2,APY3]


SET top_APY TO None


FOR num IN list1:
    if(top_APY is None or num > top_APY):
        SET top_APY TO num
OUTPUT("Highest APY:", round(100*top_APY,6),"%")


#Balance & Bank Output


SET list2 TO [B1,B2,B3]


SET best_bank TO None


FOR num IN list2:
    if(best_bank is None or num> best_bank):
        SET best_bank TO num


OUTPUT("Highest Balance: $", round(best_bank,2))
OUTPUT('Bank with the Highest APY & Balance:', str)


#Telling user bank options to choose from


OUTPUT("Bank options to choose from IN menu:")


## Reading Bank List from an External File

DEFINE FUNCTION main():
    SET file TO "banklist.txt"
    OUTPUT("\n---------------------------------\n")
    displayWithListComprehension(file)
```

```
DEFINE FUNCTION displayWithListComprehension(file):
    SET infile TO open(file, 'r')
    SET listPres TO [line.rstrip() FOR line IN infile]
    infile.close()
    OUTPUT(listPres)


main()


# Write Bank Options ($,%) & Best Bank ($,%) to text files



DEFINE FUNCTION main():
    SET outfile TO open ("banks.txt",'w')
    createWithWritelines(outfile)
    SET outfile TO open("bestbank.txt",'w')
    createWithWrite(outfile)


DEFINE FUNCTION createWithWritelines(outfile):
    outfile.writelines("Commonwealth Bank")
    outfile.writelines(":$2081, 4%\n")
    outfile.writelines("Beyond Bank")
    outfile.writelines(":$2101, 5%\n")
    outfile.writelines("St George Bank")
    outfile.writelines(":$2060, 3%\n")
    outfile.close()
```

```
DEFINE FUNCTION createWithWrite(outfile):
    SET outfile TO open("bestbank.txt", 'a')
    outfile.write("Best Bank Option\n")
    outfile.write("Beyond Bank:$2101, 5%")


    outfile.close()

main()


# Menu


DEFINE FUNCTION OUTPUTMenu():
    OUTPUT("\n Bank APY & Balance Calculator: Highest % & $")
    OUTPUT()
    OUTPUT("A: Commonwealth Bank")
    OUTPUT("B: Beyond Bank")
    OUTPUT("C: St George Bank")
    OUTPUT("X: Exit")


 # Retrieving Data Function


DEFINE FUNCTION getData1():
    SET numbersList TO [B1,100*APY1]
    OUTPUT('Balance ($) & APY (%):', numbersList)
    numbersList.append(int(INPUT()))
    RETURN numbersList
```

```
DEFINE FUNCTION getData2():

    SET numbersList TO [B2,100*APY2]

    OUTPUT('Balance ($) & APY (%):', numbersList)

    numbersList.append(int(INPUT()))

    RETURN numbersList


DEFINE FUNCTION getData3():

    numbersList =[B3,100*APY3]

    OUTPUT('Balance ($) & APY (%):', numbersList)

    numbersList.append(int(INPUT()))

    RETURN numbersList


main()

## Main Menu Function


DEFINE FUNCTION main():

        OUTPUTMenu() # Loading the menu.

        WHILE True:

            SET choice TO INPUT ().upper()

            IF choice EQUALS "O":

                OUTPUTMenu()

            ELSEIF choice EQUALS "A":

                OUTPUT("Commmonwealth Bank")

                SET myData1 TO getData1() # Getting values FOR Commonwealth Bank

                OUTPUT(myData1)
        ELSEIF choice EQUALS "B":
            OUTPUT("Beyond Bank")
            SET myData2 TO getData2() # Getting values FOR Beyond Bank
            OUTPUT(myData2)
        ELSEIF choice EQUALS "C":
            OUTPUT("St George Bank")
            SET myData3 TO getData3() # Getting values FOR St George Bank
            OUTPUT(myData3)
        ELSEIF choice EQUALS "X":
            exit()
        ELSE:
            break
            OUTPUT("\nO: Display Program options\n\n")

main()
```

8. Code your solution using Python-based on step 7 above.

[MARKS: 55marks for including below elements]

```
## Date last changed:8/04/2022
## Author: Alex Greenfield
## Date created: 15/03/2022
## This program is an APY calculator, comparing 3 different banks.
## Inputs: r, m, P, Output: APY % & Balance $
## Balance ($) & APY (%) Calculator: Commonwealth Bank (1), Beyond Bank (2), St George(3)
## Read Input: banklist.txt, Write Outputs: bestbank.txt, banks.txt



COMMONWEALTH_BANK = "Commonwealth Bank"
BEYOND_BANK = "Beyond Bank"
STGEORGE_BANK = "St George Bank"


#Bank1


Question1 = input("Is interest being compounded for atleast 1 year?(True/False):")
if Question1 == "True":
    boolean = True
elif Question1 == "False":
    boolean = False
else:
    boolean = "Error"
    print("Please input (True/False)?")

if boolean == True:
    T=float(input("How many years will the interest be compounded?-"))
    P=(float(input("Insert Principal $-")))

    r1=(float(input("Insert Commbank interest rate=")))
    m1=(float(input("Insert Commbank Compound Interval=")))


#B1 = p*(1+r1/m1)**m1

#APY1 = (1+r1/m1)**m1-1


t = 0


while t<= T:
        B1 = P*(1+r1/m1)**m1
        APY1 = (1+r1/m1)**m1-1
        t = t+ 1
        print("year =", t)
        print("$", B1)
        print("%", 100*APY1)
        break



#Bank2



Question2 = input("Is interest being compounded for atleast 1 year?(True/False):")
if  Question2 == "True":
     boolean = True
elif Question2 == "False":
     boolean = False
else:
    boolean = "Error"
    print("Please input (True/False)?")

if boolean == True:
    T =float(input("How many years will the interest be compounded?-"))
    P =(float(input("Insert Principal $-")))

    r2=(float(input("Insert Beyond Bank interest rate=")))
    m2=(float(input("Insert Beyond Bank Compound Interval=")))


#B2 = p*(1+r2/m2)**m2

#APY2 = (1+r2/m2)**m2-1




    t = 0

while t<= T:
        B2 = P*(1+r2/m2)**m2
        APY2 = (1+r2/m2)**m2-1
        t = t+ 1
        print("year =", t)

        print("$", B2)
        print("%", 100*APY2)
        break

str = 'Beyond Bank'



#Bank3
```

```
Question3 = input("Is interest being compounded for atleast 1 year?(True/False):")
if  Question3 == "True":
     boolean = True
elif Question3 == "False":
    boolean = False
else:
    boolean = "Error"
    print("Please input (True/False)?")

if boolean == True:
    T =float(input("How many years will the interest be compounded?-"))
    P =(float(input("Insert Principal $-")))

    r3=(float(input("Insert St George interest rate=")))
    m3=(float(input("Insert St George Bank Compound Interval=")))


#B3 = p*(1+r3/m3)**m3

#APY3 = (1+r3/m3)**m3-1



    t = 0

while t<= T:
        B3 = P*(1+r3/m3)**m3
        APY3 = (1+r3/m3)**m3-1
        t = t+ 1
        print("year =", t)

        print("$", B3)
        print("%", 100*APY3)
        break


#APY Output

list1 = [APY1,APY2,APY3]

top_APY = None

for num in list1:
     if(top_APY is None or num > top_APY):
        top_APY = num
print("Highest APY:", round(100*top_APY,6),"%")


#Balance & Bank Output
```

```
list2 = [B1,B2,B3]

best_bank = None

for num in list2:
    if(best_bank is None or num> best_bank):
       best_bank = num

print("Highest Balance: $", round(best_bank,2))
print('Bank with the Highest APY & Balance:', str)


#Telling user bank options to choose from


print("Bank options to choose from in menu:")




## Reading Bank List from an External File


def main():
    file = "banklist.txt"
    print("\n-----------------------------------\n")
    displayWithListComprehension(file)


def displayWithListComprehension(file):
    infile = open(file, 'r')
    listPres = [line.rstrip() for line in infile]
    infile.close()
    print(listPres)


main()



# Write Bank Options ($,%) & Best Bank ($,%) to text files



def main():
    outfile = open ("banks.txt",'w')
    createWithWritelines(outfile)
    outfile = open("bestbank.txt",'w')
    createWithWrite(outfile)

def createWithWritelines(outfile):
    outfile.writelines("Commonwealth Bank")
    outfile.writelines(":$2081, 4%\n")
    outfile.writelines("Beyond Bank")
    outfile.writelines(":$2101, 5%\n")
    outfile.writelines("St George Bank")
    outfile.writelines(":$2060, 3%\n")
    outfile.close()


def createWithWrite(outfile):
    outfile = open("bestbank.txt", 'a')
    outfile.write("Best Bank Option\n")
    outfile.write("Beyond Bank:$2101, 5%")

    outfile.close()

main()




# Menu



def printMenu():
    print("\n Bank APY & Balance Calculator: Highest % & $")
    print()
    print("A: Commonwealth Bank")
    print("B: Beyond Bank")
    print("C: St George Bank")
    print("X: Exit")



 # Retrieving Data Function



def getData1():
    numbersList = [B1,100*APY1]
    print('Balance ($) & APY (%):', numbersList)
    numbersList.append(int(input()))
    return numbersList

def getData2():
    numbersList = [B2,100*APY2]
    print('Balance ($) & APY (%):', numbersList)
    numbersList.append(int(input()))
    return numbersList

def getData3():
    numbersList =[B3,100*APY3]
    print('Balance ($) & APY (%):', numbersList)
    numbersList.append(int(input()))
    return numbersList

main()



## Main Menu Function



def main():
        printMenu() # Loading the menu.
        while True:
            choice = input ().upper()
            if choice == "O":
                printMenu()
            elif choice == "A":
                print("Commmonwealth Bank")
                myData1 = getData1() # Getting values for Commonwealth Bank
                print(myData1)
            elif choice == "B":
                print("Beyond Bank")
                myData2 = getData2() # Getting values for Beyond Bank
                print(myData2)
            elif choice == "C":
                print("St George Bank")
                myData3 = getData3() # Getting values for St George Bank
                print(myData3)
            elif choice == "X":
                exit()
            else:
                break
                print("\nO: Display Program options\n\n")

main()
```