**EXERCICE 1 :**

On souhaite créer une application Python permettant la gestion des comptes bancaires dans une banque :

* Un compte bancaire est défini par un identifiant unique du titulaire, le numéro de compte et son solde
* Deux opérations de base sont possibles sur son compte : déposer et retirer de l’argent
* Aucune facilité de caisse n’est autorisée (i.e. le solde d’un compte ne peut pas être négatif)
* Les acteurs de cette application sont : l’agent de la banque et un client. Chaque client dispose d’un code secret pour accéder à son compte. On suppose que chaque client a un seul compte

**Nb:**

• le numéro du client est incrémental

• Le numéro de compte est un entier formé par le numéro de son propriétaire et un nombre aléatoire compris entre 0 et 100

Exemple : Si le numéro du client est 5 alors son numéro de compte est par exemple : **556** (où **5** est le numéro du client et **56** est un nombre aléatoire compris entre 0 et 100)

🡪 A travers un menu de choix, cette application offre à l’agent les fonctionnalités suivantes :

1. Ajouter un Compte

2. Supprimer un Compte

3- ENREGESTREMENT

🡪A travers un menu de choix cette application offre à un client les fonctionnalités suivantes:

1. Modifier son mot de passe

2. Afficher son solde

3. Déposer une somme d’argent

4. Retirer une somme d’argent

**Indication** : Utiliser la structure de données dictionnaire de Python pour la gestion des comptes et des clients

**Exemple :**

•Le dictionnaire **Compte** associe à chaque numéro de Compte son solde

Exemple : Compte={56:200, 20:360} signifie que le solde du compte numéro 56 est 200Dh et le solde du compte numéro 20 est 360Dh

•Le dictionnaire **Client** qui associe à chaque numéro de client son code secret

Exemple : Client = {1:’’566’’, 2:’’836’’} signifie que le code secret du client numéro 1 est égal à 566 et le code secret du client numéro 2 est égal à 836

•Le dictionnaire **ClientCompte** associe à chaque numéro de client son numéro de compte

Exemple : ClientCompte{56:1,20:2} signifie que le numéro de compte du client 56 est 1et le numéro de compte du client 20 est 2

TRAVAIL A FAIRE :

1. On considère Client, Compte et ClientCompte les dictionnaires utilisés pour la gestion des comptes. Définir les fonctions suivantes relatives à l’agent de la banque :

* **ajouterClient**( numCl, MPC, numC, SoldeC) tels que: numCl est le numéro du nouveau client, MPC est son code secret, numC est le numéro de son compte et SoldeC est le solde initial de son compte
* **SupprimerClient(numC)** tel que numC est le numéro du compte à supprimer

1. Définir les fonctions suivantes relatives à un client de la banque (identifier les paramètres de ces fonctions)

* **Modifier MPClient**: permettant de modifier le mot de passe d’un client
* **Déposer :** permettant de déposer une somme d’argent dans un compte
* **Retirer :** permettant de retirer une somme d’argent d’un compte

1. Définir la fonction lambda genererNumCompte permettant de générer le numéro de compte d’un client à partir du numéro de ce client
2. Ecrire une fonction EcrireFichierCSV permettant d’enregistrer les numéros des clients et leurs codes secrets dans un fichier CSV
3. Ecrire une fonction manipSTS permettant de créer une liste, un tuple et un set de numéro de comptes à partir du dictionnaire *ClientCompte*
4. Ecrire un programme principal permettant de tester les fonctions définies précédemment. Proposer un menu de choix à l’agent et aux clients de la banque

**Remarques :**

🡪La fonction ***modifierMPClient (numCl, anciemMP, nouveauMP)*** ayant les paramètres suivants:

•**numCl**: numéro du client souhaitant changer son mot de passe

•**ancienMP**: son ancien mot de passe.

•**NouveauMP**: son nouveau mot de passe.

🡪La fonction ***deposer ( NumCl, soldeD)*** ayant les paramètres suivants :

•**NumCl**le numéro de client souhaitant déposer de l’agent

•**soldeD**: le solde à déposer

🡪La fonction ***retirer ( NumCl, soldeR)*** les paramètres suivants :

•**NumCl**le numéro de client souhaitant déposer de l’agent

•**soldeR**: le solde à retirer

•Rappeler le fait qu’il est impératif de vérifier si le numéro du client existe et si le montant retiré est autorisé

def ajouterClient(numCl, MPC, numC, SoldeC):  
 return True  
def SupprimerClient(numC):

return True  
def MPClient(MPC):  
 return True  
def Deposer(NumCl, soldeD):  
 return True  
def retirer ( NumCl, soldeR) :  
 return True  
def modifierMPClient (numCl, anciemMP, nouveauMP) :  
 return True  
def ecrireFichierCsv():  
 return True  
def modifierMPClient(numCl, anciemMP, nouveauMP):  
  
 Client[numCl]=anciemMP  
 Client[numCl]=nouveauMP  
   
def Deposer(NumCl, soldeD):  
 compte[int(lambdanumC(NumCl))]+=soldeD  
  
def retirer ( NumCl, soldeR) :  
 compte[int(lambdanumC(NumCl))] -= soldeR  
  
  
lambdanumCompte=lambda c:c#num de compte à générer  
#programme principal  
numCl=4  
Client = {1: '556', 2: '889', 3: '446', 4: '223'}  
Compte = {10: 200, 20: 360, 30: 460, 40:560}  
ClientCompte = {1: 10, 2: 20, 3: 30, 4:40}  
choixAC=int(input("""si vous êtes agent taper 1  
si vous êtes client taper 2   
taper votre choix : """))  
if choixAC==1:#vérification agent  
 mp=input('donner votre mot de passe :')  
 if mp=='1234':  
 continuer=True  
 while continuer==True:  
 print("\*\*\*\*\*\*\*\*\*")  
 print("1-ajouter un compte client:")  
 print("2-supprimer un compte client:")  
 print("3-generer un compte client:")  
 choix=False  
 while choix==False:  
 numchoix=input('taper le numero de votre choix :')  
 if numchoix=='1':  
 choix=True  
 #ajouter  
 numCl=numCl+1  
 numC=lambdanumCompte(numCl)  
 solde=input('donner le solde :')  
 ajouterClient(numCl,'23',int(numC),int(solde))  
 print('client ajouter avec succés')  
 elif numchoix=='2':  
 choix=True  
 #supprimer  
 numClsup=input('donner le numero de client :')  
 SupprimerClient(int(numClsup))  
 print('client supprimer avec succés')  
 elif numchoix=='3':  
 choix=True  
 #écrire fichier  
 ecrireFichierCsv()  
 print('client générer avec succés')  
 else:  
 print('taper 1 ou 2 ou 3 :')  
 rep=input('voulez vous continue oui ou non:')  
  
 if rep=='non':  
 continuer=False  
 else:  
 print('mot de passe incorect')  
elif choixAC==2:  
 choixNumCl=False  
 while choixNumCl==False:  
 X=input('taper votre numero \n 1 Modifier son mot de passe \n 2 Afficher son solde \n 3 Déposer une somme d’argent \n 4 Retirer une somme d’argent ')  
 if choixNumCl=="1":  
 choixNumCl==True  
 numCl = int(input('taper numCl'))  
 anciemMP = input('taper votre ancien Mot de Passe:')  
 nouveauMP = input('taper votre nouveau Mot de Passe:')  
 print(modifierMPClient(numCl, anciemMP, nouveauMP))  
 elif choixNumCl == 2:  
 ClChoice = True  
 numCl = int(input('taper numCl'))  
 numC = lambdanumC(numCl)  
 print(compte[int(numC)])  
 elif choixNumCl == 3:  
 ClChoice = True  
 soldeD = int(input('taper le solde à deposer :'))  
 numCl = int(input('taper numCl'))  
 print(Deposer(numCl, soldeD))  
 elif choixNumCl == 4:  
 ClChoice = True  
 soldeR = int(input('taper le solde à retirer :'))  
 numCl = int(input('taper numCl'))  
 if soldeR >= compte[int(lambdanumC(NumCl))]:  
 retirer(numCl, soldeR)  
 else:  
 print('erreur ')  
 else:  
 print('taper 1 , 2 ,3 ou 4')  
 rep = input('voulez vous continue ou non :')  
 if rep.lower() == 'oui':  
 ClChoice = False  
 else:  
 break

*"""  
verefication recherche client  
3 dictionner  
  
"""*def menu():  
 while True:  
 print("agent taper 1")  
 print("client taper 2")  
 choix = int(input("choisi le choix :"))  
 if 0 < choix < 3:  
 break  
 return choix  
def menu\_agent():  
 while True :  
 print("1. Ajouter un Compte")  
 print("2. Supprimer un Compte")  
 print("3-generer un compte client")  
 choix\_agent=int(input("choisi le choix :"))  
 if 0<choix\_agent<4:  
 break  
 return choix\_agent  
  
  
  
def ajouterClient(numCl, MPC, numC, SoldeC):  
 compte[numC]=SoldeC  
 Client[numCl]=str(MPC)  
 ClientCompte[numC]=numCl  
  
def SupprimerClient(numC):  
 del compte[numC]  
 del Client[numCl]  
 del ClientCompte[numC]  
def ecrireFichierCsv():  
 import csv  
 tp=[compte,Client,ClientCompte]  
 fichier = open("bon-commande.csv", "w")  
 csv.DictWriter(fichier, delimiter=";", fieldnames=tp[0].keys())  
  
  
def modifierMPClient(numCl, anciemMP, nouveauMP):  
  
 Client[numCl]=anciemMP  
 Client[numCl]=nouveauMP  
  
  
def Deposer(NumCl, soldeD):  
 *""" x = compt[ncl]  
  
 compte[x] = compt[x] + solde  
 print(" votre solde est ", compt[x])"""*def retirer ( NumCl, soldeR) :  
 *""" x=compt[ncl]  
  
 compte[x]=compt[x]-solde  
 print(" votre solde est ", compt[x])  
 #test solde  
 """*import random  
import math  
c=math.floor(random.uniform(10,100))  
lambdanumC=lambda x:str(numCl)+str(c)  
numCl=4  
compte={12:200,29:300,30:566,46:345}  
Client={1:'345',2:'234',4:'908'}  
ClientCompte={12:1,29:2,30:3,46:4}  
  
choix=menu()  
if choix == 1:  
 choix\_agent=menu\_agent()  
 agchoice = False  
 while agchoice == False:  
 if choix\_agent == 1:  
 agchoice = True  
 numCl = numCl + 1  
 MPC = int(input('taper votre mot de passe'))  
 numC = lambdanumC(numCl)  
 SoldeC = int(input('taper le solde :'))  
 ajouterClient(numCl, MPC, numC, SoldeC)  
 print('client ajouter avec succés')  
 elif choix\_agent == 2:  
 agchoice = True  
 numC = lambdanumC(numCl)  
 SupprimerClient(numC)  
 print('client supprimer avec succés')  
 elif choix\_agent == 3:  
 agchoice = True  
 ecrireFichierCsv()  
 print('client générer avec succés')  
 else:  
 print('taper 1 ou 2 ou 3')  
 rep = input('voulez vous continue ou non :')  
 if rep.lower() == 'oui':  
 agchoice = False  
 else:  
 break  
  
elif choix == 2:  
 clientChoix = int(input("""1. Modifier ton mot de passe  
2. Afficher ton solde  
3. Déposer une somme d’argent  
4. Retirer une somme d’argent  
taper votre choix:"""))  
 ClChoice = False  
 while ClChoice == False:  
  
 if clientChoix == 1:  
 ClChoice = True  
 numCl = int(input('taper numCl'))  
 anciemMP = input('taper votre anciem Mot de Passe:')  
 nouveauMP = input('taper votre nouveau Mot de Passe:')  
 print(modifierMPClient(numCl, anciemMP, nouveauMP))  
 elif choix == 2:  
 ClChoice = True  
 numCl = int(input('taper numCl'))  
 numC = lambdanumC(numCl)  
 print(compte[int(numC)])  
 elif choix == 3:  
 ClChoice = True  
 soldeD = int(input('taper le solde à deposer :'))  
 numCl = int(input('taper numCl'))  
 print(Deposer(numCl, soldeD))  
 elif choix == 4:  
 ClChoice = True  
 soldeR = int(input('taper le solde à retirer :'))  
 numCl = int(input('taper numCl'))  
 if soldeR >= compte[int(lambdanumC(NumCl))]:  
 retirer(numCl, soldeR)  
 else:  
 print('erreur ')  
 else:  
 print('taper 1 , 2 ,3 ou 4')  
 rep = input('voulez vous continue ou non :')  
 if rep.lower() == 'oui':  
 ClChoice = False  
 else:  
 break

import random

import os

import csv

def manipSTS():

liste=[]

for v in Client.values():

liste.append(v)

tup=tuple(liste)

sete=set(liste)

def EcrireFichier():

file=open("fich.csv","w")

writer = csv.writer(file)

for k,v in Client.items():

writer.writerow([k,v])

file.close()

def ajouterclient(numclient,MPC, numCompte, SoldeC):

Client[numclient]=MPC

Compte[numCompte]=SoldeC

ClientCompte[numclient]=numCompte

print("Client :",Client)

print("Client :",Compte)

print("Client :",ClientCompte)

def Supprimer\_client(numc):

r=numc/10

del Client[r]

del ClientCompte[r]

del Compte[numc]

def ModifierMDP(numc,nouveau):

nouveau=input("votre nouveau mdp : ")

Client[int(numc)]=nouveau

def deposer(numc):

argent=int(input("saisir le mantant : "))

Compte[numc]+=argent

def retirer(numc):

argent=int(input("saisir le mantant : "))

if argent<=Compte[numc]:

Compte[numc]-=argent

else:

print("pas de solde")

def lambdannumcompte(numclient):

return str(numclient)+str(0)

Client = {1:'566', 2:'836',3:'456'}

Compte={10:200, 20:360,30:460}

ClientCompte={1:10,2:20,3:30}

numclient=3

while 1:

choixAC=input("si vous etes un agent taper 1,et si vous etes client taper 2:")

if choixAC=='1':

MDP=input("taper votre mot de passe:")

if MDP=="123":

continuer=True

while continuer==True:

print("\*\*\*\*\*\*menu\*\*\*\*\*\*\*:")

print("1-ajouter un compte client:")

print("2-supprimer un compte client:")

print("3-generer un compte client:")

choix=False

while choix==False:

numchoix=int(input("taper le numero de votre choix:"))

if numchoix==1:

choix=True

numclient=numclient+1

numcompte=lambdannumcompte(numclient)

solde=input("Donner le solde:")

ajouterclient(numclient,"1234",int(numcompte),int(solde))

print("client ajouter avec succes !")

elif numchoix==2:

choix=True

numclientsupp=input("donner le numero de compte a supprimer:")

Supprimer\_client(int(numclientsupp))

print("le client supprimer avec succes.")

elif numchoix==3:

choix=True

EcrireFichier()

print("fichier client generer avec succes")

else:

print("tapez 1 ou 2 ou 3:")

rep=input("voulez vous continuer?:")

if rep=="0":

continuer=False

else:

print("mot de passe incorrect:")

elif choixAC=='2':

MDP=input("entrer votre mot de passe")

if MDP == "1234":

print("1:modifier le mot de passe","\n","2:deposer","\n","3;retirer")

continuer=False

while continuer== False:

choix=input("taper votre choix : ")

if choix=='1':

continuer=True

num2client=input('saisir le numero de client : ')

ModifierMDP(num2client)

print("succes")

elif choix == '2':

continuer=True

num2client=int(input('saisir le numero de compte : '))

deposer(num2client)

print("succes")

elif choix=='3':

continuer=True

num2client=int(input('saisir le numero de compte : '))

retirer(num2client)

print("succes")

else:

print("taper 1 , 2 ou 3")

rep=input("voulez vous continuer?")

if rep =="0":

continuer=True

else:

print("mot de passe incorrect")

p=input("voulez vous continuer?")

if p =="0":

break