Module 4	COUITSETCI ① Melp	
In this assignment, you will implement an online basingry gridern. Uters can sign-up with the system, change their password, and delete their account. They can also update their bank account balance and transfer money to another user's bank account. You'll implement functions related 5F Is II and dictorance. Two of the functions require you to import files and create dictorances. User information will be imported from the "users stiff the and account information will be imported from the "transfer file. The remaining functions require you to use or modify the level dictorances created from the files. Each function has been defined for you, but without the code. See the dicciting in each functions in supposed to do and how to write the code. It should be clear enough, in some cases, we have provided their to help you get started.		
10 [1] ### ###############################		
This function is used to create a bank distributy. The given argament is the filename to load. Formy lies in the file should be in the following format: key: value The key is a coper's man amount to update the user's bank account with. The value should be a		
mader, Nonever, it is possible that there is no value or that the value is an ironical number. Shoty was will do: - Create as easy hand dictionary, - Add any and values to the dictionary, from the contents of the file. - If the key done it which is the dictionary, treate a none key value part. - The key done is a simple for following cases: - The value of the dictionary of the dictionary, treate a none for the dictionary. - The distribution of the dictionary of the dictionary, treated the dictionary. - The distribution of the dictionary of the dictionary. - The distribution of the dictionary of the dictionary. - The distribution of the dictionary of the distribution of the dictionary. - The distribution of the dictionary from this distribution of a line and only update the dictionary. - The distribution of the distribution of the dictionary of the distribution of the dictionary of the distribution of the dictionary. - The distribution of t		
when the value is sixing or invalid. If \$\(\text{s}_i \) (spece that it lies and don't update the dictionary. when the lies is completely blank. again, againer that it can do not update the dictionary. should tris any and all addrespace. bettern the beam dictionary from this function. for example, here's been your code should headle uses specific lies in the file: but it is to the lift has a meam and valid andmander.		
Two code will process this line and add the extracted information to the dictionary. After it does, the dictionary will look like this: bank "(Paradont's)		
The high lines in the file also has a came and valid number: Partia: 18.7 Your code will also process title line and add the extracted information to the dictionary. After it does, the dictionary will look like this: See the file of the file o		
The 4th line in the file has a name but sizzing number:		
Jacob control for the file is a desired to the dictionary. It will still look like this: Described to the file is completely thous. So this is the file is completely thous. The file is the file is completely thous. The file is the file is completely thous. The file is the file is the file is completely thous. The file is the file is the file is a none outlin conter, but with extra addrespace: The file is the file is a none outlin conter, but with extra addrespace: The content is the file is a none outlin conter, but the existing key ('Benedon') in the dictionary. After it does, the value associated with the key 'Benedon' will be all: The file is the file is the substantial to the let is the file		
After it does, the value associated with the key 'ferendor' will be 18: ban' ("Frendor' 15: "		
bask - () # your code here with open(Thineaux, 'r') as file: for line in file:		
with quoted (lines, "") as file: for the in file:		
auti-mane(1).trip() # using infigit() * replace() # Check for Finds rating rea - wait.replace(', '', ', ', ', ').indight() # print routil		
res = relt.replace(, ', ') issispit() # print relation # rest		
# 1845 # 2004		
## ## ## ## ## ## ## ## ## ## ## ## ##		
In []: internet and the control of t		
Tools.asser_laber_epoil(20.87). bear_ept("broken"); Tools.asser_laber_epoil(20.87). bear_ept("broken"); Tools.asser_laber_epoil(20.87). bear_ept("broken"); Tools.asser_laber_epoil(20.87). bear_ept("broken"); Tools.asser_laber_epoil(20.87). ept("broken"); Tools.asse		
In []: def signus(nese_accounts, log_is, username, passenrd): This function allows users to sign up. If both sucremes and passenram and passenred must the requirements: passenreme passenreme passenreme passenrement		
- Between Part State Control of State Co		
- The passaned must contain at least one masher. The acceptage agreement passaned must be the same. For example: - Colling signaptors accounts, log_in, "Brandom", "123AdVACO") will return false - Colling signaptors accounts, log_in, "Rendom", "123AdVACO", will return false - Colling signaptors accounts, log_in, "mandom", "123AdVACO", "123AdVACO" again will return false - Colling signaptors accounts, log_in, "fraudom", "123AdVACO" again will return false. - Colling signaptors accounts, log_in, "fraudom", "123AdVCO" again will return false.		
Hist: Thick about defaults and a single a separate valid(passeows) function that checks the validity of a given passeord. This will lais cause in beauty and survey the separate valid passeord function.		
# your code here # Guernam mot E user_scounts: * result * usilizate(username, passourd) print(result) # Fregult = Trus: # add to Sitt		
# years mean the user_accounts; result = validate(usersume, passumed) print(result) # result = True: user_account(usersum) = passumed # set of true; # result = True; # set of true; #		
esters result		
In []: (def import_and_create_accounts(filenems): This function is used to create an user accounts dictionary and another logic dictionary. The given argument is the filenems to load.		
This function is used to create an user accounts dictionary and another login dictionary. The given argument is the fileness to load. File should be in the following format: File function are assumed. The product of		
For the logic dictionary, the boy is the unersease, and its value indicates whether the user is legged in, or not. Sitting, all user are one logged in. What you will de: - Create an early user accounts dictionary and an empty legic dictionary. - Read in the file. - Read in the file. - Read in the file. - The sear accounts dictionary and grant for file in the requirement, asks the increase and parameter - Interest the sear accounts discinstancy, and updates the logic dictionary. - You bound also houside the following creati - What there is subletges at the Registration of a file analyse between the name and parameter - What there is subletges at the Registration of a file analyse between the name and parameter of a line and/or between the name and parameter when the user accounts discinstancy from this function.		
- You should also handle the following cause: - when they passed is sintle, I is o, ignore that line and don't update the dictionaries when there is whitespice at the beginning or end of a line and/or between the mean and passeured on a line. You should rise appear all uniterpace Neture both the user accounts dictionary and login dictionary from this faction For example, here's how you code should headed see specific! lines in the file:		
For example, here's hew your case should headle stems specific lines in the fills to be little into the fills have a seame and passered. Broadles branches Mark I do as a seame and passered in the structed information to the distinction. Set of the structure of the structure. After it does not set distinctions will look like this: star seconds - (Twendom' Twendom')Addr() Leg is - (Twendom' Twendom')Addr()		
The 2 miles for the file has a same but mixing password: The control of the file has a same but mixing password: The control of the file has a same but mixing a same and a same a sam		
The lot line in the file has a mane and parametr: Your codes will process this line, and waige the signor function, will not add the extracted information to the Grant of the state of t		
The 481 lies to the file but a name and partnered: Now code will process this line, and using the singue function, will add the extracted information to the discharge. As of present visit line, and using the singue function, will add the extracted information to the discharge. As of present visit is a singular visit of the sing		
Miss processing every like in the Gila, the dictioneries will look like this user scenaries ("window) "restandings," planting "restandings," planting "restandings," planting "restandings," planting "restandings," planting the distinguishers the distinguishers we this fraction. were accounts of the distinguishers with fraction. were accounts of the distinguishers with fraction.		
# pair code here value=0 # pair code here		
for line is fill; prior() name 'line 'risting'''), quilt("-") #imane, ont) - (ine 'risting'''), quilt(")'')		
### (10 (10 (10 (10 (10 (10 (10 (10 (10 (10		
resum lidates (minos), passanera) if res: #print (passaner) value-passaner #klet #print False		
# 1.45		
### saper common super class (buy : value)) log_in.update(buy : "false")) ### if it is not common super class (buy : "false")) ### if it is not common super class (buy : "false")		
result = false for in assumed: # constraint quarteres alphabets if (i.i.slaw()); # counting approvas alphabets if (i.i.super());		
return varie_score() inc. [in.] de visit (secretary = 1) in the control of the c		
result = false		
Section Sect		
tools assert the first the		
tools.ssert_false(signo/user_accounts, log_in_"branch", "l23abcb60"))		
tools.sserg_tales(impro(our_accounts.ing.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in_transcorrents.in		
tools.asserlrus(isjump(oseaccounts, lagin*Trandost_*_*_TllassCD')) tools.asserclus(isjump(oseaccounts_ing_in*Trandost_*_*_TllassCD')) tools.asserclus(isjump(oseaccounts_in_*_*_TllassCD')) tools.asserclus(isjump(oseaccounts_in_*_*_TllassCD')) tools.asserclus(isjump(oseaccounts_in_*_*_TllassCD')) tools.asserclus(isjump(oseaccounts_in_*_*_TllassCD')) tools.asserclus(isjump(oseaccounts_in_*_*_TllassCD')) tools.asserclus(isjump(oseaccounts_in_*_*_TllassCD'))		
This function allows users to leg in with their variences and parament. The way, contest distribute varience the unermose descrized parament. The lag_is distributely views the variences and saterized lag in twice. If the varience does not exist it near generate the paramed is increased.		
Otherwise - Updates the user's lag-in status in the lag in dictionary, setting the value to True Autoris True For example: - Calling lagsin/clear_accounts, "femandum", "lalakeda") will return False - Calling lagsin/clear_accounts, "meandor", "meandorilaked") will return True		
# your code have		
In [] SPRENEWARDERSERVER OF IST TOUR SOLUTION OF THE TOUR SOLUT		
print("Second") In [1] (de quadratina, jag ja, warrama, amount) In [1] (de quadratina, you will my to quadra the given user's host account with the given amount. In [1] (de print 1 + 2 decision) you will make the given user's host account with the given amount. In [1] (de print 1 + 2 decision) you will not be a given by a given with a given amount. In [1] (de print 1 + 2 decision) you will not be a given by a given with a given amount. In [1] (de print 1 + 2 decision) you will not be a given with a given amount. In [1] (de print 1 + 2 decision) you will not be a given with a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [1] (de print 1 + 2 decision) you will not be a given amount. In [
log_fo is a distinsive where the key is the username and the value is the user's larged real. Sometic the amount to uniform action, and can eliminate most, the following requirements must be most: To update the user's account with the amount, the following requirements must be most: The user excitation languary and mixture trains in From, asseming, the user languaged in. If the user descrit sixt in the bank, create the user. The green amount can not that in a segurity behavior in the bank account.		
- The given amount can not result in a segentive behavior in the bask account. Statem The Tiff the ware's account was applicable. For exemple, if Permation has 151.00 in this account! - Calling quinter(main, lag, 10, "Terminor", 00) will return False, unless "Srandom" is first lagged in. Then it - Calling quinter(main, lag, 10, "Terminor", 200) will return false because Brandom does not have enough in his account.		
8 year code here		
In [] constructions and the construction of t		
Total, assert_clair_colorate_loam_ing_man_ing_		
signo(ser_accounts, log_lo, "Promoter") total.serer_i_sered_i_		
In this fraction, you will ly to make a creative fraction butter account to allow a distinction years to be significant to the		
what you will do: - Ownert the given amount from userA and add it to users, which makes a transfer. - You whould consider user following class: - wind wheat to be in dig, the regardeds of liquid citates, user can be absent in the bank to user can have a magnitum amount in their account. Works an extinct or zero bilance No user can have a magnitum amount in their account. Works much have a positive or zero bilance.		
For example: - Gilling transfer(base, log_in, "broadout", "laid", 180) will return false - Gilling transfer(base, log_in, "broadout", "laid", 180) will return false - After logging "broadout", cilling transfer(base, log_in, "broadout", "laid", 'laid", 'laid vill return True - Gilling transfer(base, log_in, "broadout", "laid", 'laid", 'laid vill return false		
# your code here To [1: Immunimummum Immunimummum Immunimummum Immunimum		
In [] :: :: :: : : : : : : : : : : : : :		
ingin_tear_act_act_act_act_act_act_act_act_act_act		
Signation accounts, lag. As "grounders" Tabladary		
In [] left change_assumer(cleer_ascounts_in_change_ascounts_in_change_ascount); This is never their changes ascounts_in_change in the change i		
If all of the following requirements are met, changes the password and returns True. Otherwise, returns false. The suremann exists in the user_accounts. The same is lagged in the suremann is associated with the value True in the lag_in dictionary) The man_password following requirements from the all down. The man_password following recounts. Joes in the requirement in signor. For example: Called those measurements are also "recounts, and in. "recolors" "liblacker" "liblacker".		
For exception - Calling change paramenf(user_accounts, log_in, "Rendomer", "123accARC", "123accARC") will return false - Calling change paramenf(user_accounts, log_in, "Rendom', "Armonia Table ("123accARC") will return false - Calling change paramenf(user_accounts, log_in, "Rendom', "Paramenf ("124accounts, "vanabounts) will return false - Calling change paramenf(user_accounts, log_in, "Rendom', "Paramenf ("124accounts, "vanabounts) will return false HHH: Thick about defailing and using a separation language of Armonia Table ("123accounts that checks the welldity of a given parament. This will also come in heady when writing the signost) function that checks the welldity of a given parament.		
# your code here In [] ##################################		
bas = 16per_sat_ceres_bas("shar_sat_cert_) tost_state("shar_sat_cert_) tost_state("sha		
tost.ssser_slate(cheap_seasone(us_srconts_ing_in_"newnort_state(")) tost.ssser_slate(cheap_seasone(us_srconts_ing_in_"newnort_state(")) tost.ssser_slate(cheap_seasone(us_srconts_ing_in_"newnort_state("))		
tolia assett Turoficheng parametriforer, accounts, ing fu, "avadent" "for admental 2002", "123 anc. ACCO")) (roll assett Turofichens) [2] [2] (der dalete, account(une_accounts, ing fu, "avadents, "ing accounts, "ing fu, "avadents, "ing accounts, "ing fu, accounts, "ing fundamental account		
Completely deletes the user from the notice heating system. If the user exists in the user_accounts dictionary and the parameter is legged in (the username is associated with the value True in the log_in dictionary): - Notices the user_accounts dictionary and the heat dictionary, and the heat dictionary, and the heat dictionary and the heat dictionary. - Otherwiser - Notices (Alexander accounts los to heat "Research" "Tileschell") will content folia.		
for exemple: - Gilling Saltes_account(uner_accounts, log_in, hash, "francher", "123accast() will return false - Gilling Saltes_account(uner_accounts, log_in, hash, "breacher", "123accast() will return false - Gilling Saltes_account(uner_accounts, log_in, hash, "breacher", "123accast() will return false - Will return True "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher"), "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher"), "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_account(nerr_accounts, log_in, hash, "Breacher") - Wronder ", "Breacher" in, calling Saltes_accounts, log_in, hash, "Breacher" - Wronder ", "Breacher" in, calling Saltes_accounts, log_in, hash, "Breacher" - Wronder ", "Breacher" in, calling Saltes_accounts, log_in, hash, "Breacher" - Wronder ", "Breacher" in, calling Saltes_accounts, log_in, hash, "Breacher" - Wronder ", "Breacher" in, calling Saltes_accounts, log_in, hash, "Breacher" - Wronder ", "Breacher" in, call		
has - Imper_and_creat_, June ("name_art", June ("name_art") vor_arcord, June ("name_art", June ("name_art") total_saser_flain(shirt, According or sacront, June (a.h.an), "personds" ("liberation") total_saser_flain(shirt, According or sacront, June (a.h.an), "personds" ("liberation") total_saser_flain(shirt, June (a.h.an), "personds ("liberation") total_saser_flain(shirt, According or sacront, June (a.h.an), "personds ("liberation") total_saser_flain(shirt, According or "personds (liberation")) total_saser_flain(shirt, June (a.h.an), "personds (liberation)) total_saser_flain(shirt, June (a.h.an), "pers		
1001.4.sservac(detaccount_tearcoun		
In [] def main(): The main function is a skeleton for you to test if your neverall programming is sarring. Note we will not test your main function. It is only for you to run and interact with your program.		
bank - Import_mod_create_bank("bank_tax") sorp_accounts_ law_be = spore_mod_create_baccounts("sare_tax") shile Tens: # for debugging print(lawal; bank) print("bank_tax) print("bank_tax) print("bank_tax) print("bank_tax)		
option = input("what do you want to do) Please enter a numerical option below.'w" "1. legislu"		
"2. signos/" "3. charge parametry" "5. update assembly" "6. update assembly" "7. exity" "7. exity" "1. university of the stername of the stern		
If option = "1": unremain = Imput ("Plains Input the internation") passerd = Imput ("Plains Input the passereditor") # and cont to large in a control in a con		
passand - import ("Passas Injust the passands)") # ond cost to right signup(one pacounts, log_fo, unersone, passands) salf option - "")" ("passa injust the cost passands)") salf_passand - import("Passas injust the cost passands)") salf_passand - import("Passas injust the cost passands)")		
# did colt to change password control to the change password c		
# deld code to delete account (see gin, land, username, passured) #16 (pution - "5"; " ("Stated input the storcurs)") #25 (see gin) #25 (see g		
secont - flact(posett) # add code to gaster amount equater(pas, log_in, username, asount) prior(The amount is localid. Please ventore the option(n')		
all egita or "t"; such - logot("Values logot the user who will be dedected or ") such - logot("Values logot the user who will be added or ") amount = logot("Values logot the amount(or") try: amount = logot("Values logot the amount(or")		
# add code to trousper mount transfer(man, [git, words, store, amount) priori("The amount is invalid. Please re-mater the option, b/") all equits """!		
the state of the s		