Import json

FUNTION Load transaction()

Try:

Open transaction.json file in read mode as a file

DOFOR transaction in transactions file

Add transaction to the end of the transactions

ENDFOR

EXCEPT ValueError:

pass

ENDFUNTION

FUNTION save transaction()

Open transaction.json file in write mode

Save the added transaction

ENDFUNTION

FUNTION add transaction()

Display add transaction

TRY:

Prompt for the amount,purpose of transaction

DOWHILE True:

about of the transaction is that a Income or Expense in capitalize() store in a about variable

DOIF about in capitalize()

BREAK

ELSE

Prompt successful message

ENDIF

DOWHILE True:

DOIF length of the date = 10

BREAK

ELSE:

Display Enter the date in correct format

ENDIF

ENDWHILE

Prompt for the date of the transaction

Get the transaction details from the user as input and store them in variables

APPEND transaction to transactions

Save transactions to the

Return the value

EXCEPT ValueError:

Display error message

ENDFUNTION

FUNTION view transaction()

Total transaction

DOIF no transaction in transactions

Prompt “You haven't add any transactions yet.”

ELSE

Set count = 0

Prompt count amount,description,income or expense and the date

Add 1 to the count

ENDIF

ENDFUNTION

FUNTION update transaction()

Prompt for the transaction number needed to be updated

-1 from the user input

DOIF 0 < user input <= length of the transaction

Prompt for the user to select the neither amount , decription,income/expense or the date

DOWHILE True:

TRY:

Ask the selection number from the above and store it in selection variable

DOIF selection = 1

Prompt for the user for new amount

Add the new amount to the transaction[]

Save the change by calling the save\_transactions()

ELIF selection = 2

Prompt for the user for new description

Add the new description to the transaction[]

Save the change by calling the save\_transactions()

ELIF selection = 3

Prompt for the user for new about

Add the new about to the transaction[]

Save the change by calling the save\_transactions()

ELIF selection = 4

Prompt for the user for new date

Add the new date to the transaction[]

Save the change by calling the save\_transactions()

BREAK

ENDIF

EXCEPT ValueError:

Display error message

ENDFUNTION

FUNTION delete transaction()  
 prompt for the user for transaction to be deleted

Store it in a variable d\_transaction

DOIF 0 < d\_transaction <= length of the transaction

Delete the transaction in the transactions

Call save\_transactions() to save the change

ELSE

Display the “Enter a valid transaction”

ENDIF

ENDFUNTION

FUNTION display\_summary()

Display the total transaction done by the user

Assume income total as zero

Assume expense total to zero

DOFOR transaction in transactions

DOIF transaction[2] = Income

Add total income to income\_total

Display income\_total

ELSE

Add total expense to expense\_total

Display expense\_total

ENDIF

net\_total = income\_total - expense\_total

display net\_total

ENDFOR

ENDFUNTION