



In [2]:

```
pip install tabulate
```

Requirement already satisfied: tabulate in c:\users\user\anaconda3\lib\site-packages (0.8.7)  
Note: you may need to restart the kernel to use updated packages.

In [3]:

```
from tabulate import tabulate
```

In [4]:

```
from tkinter import *
```

In [5]:

```
#datastores = { "accountdetails":["token": 22331,"accountnumber": 100,"name": "reception","mail":  
50,"balance": 75,"nominee":"df"]}  
datastore = { "token": 22331,"token":22335,"token":22339,"token":22346,"token":223389}
```

In [6]:

```
class ValueTooSmallError(Exception):  
    def __init__(self, m):  
        self.message = m  
    def __str__(self):  
        return self.message  
  
class ValueTooLargeError(Exception):  
    """Raised when the input value is too large"""  
    def __init__(self, m):  
        self.message = m  
    def __str__(self):  
        return self.message  
  
class NotNumeralError(Exception):  
    """Raised if input is numeral"""  
    def __init__(self, m):  
        self.message = m
```

```

    self.message = m

    def __str__(self):
        return self.message

class SpecialSymbolError(Exception):
    ''' Raised if special symbols occur at input'''
    def __init__(self, m):
        self.message = m
    def __str__(self):
        return self.message

class UpperCaseSymbolError(Exception):
    '''Raised at the exception when a uppercase letter isnt '''
    def __init__(self, m):
        self.message = m
    def __str__(self):
        return self.message

class LowerCaseSymbolError(Exception):
    '''Raised at the exception when a lowercase letter isnt'''
    def __init__(self, m):
        self.message = m
    def __str__(self):
        return self.message

class NonAlphabetError(Exception):
    '''Raised,when Alphabet occurs'''
    def __init__(self, m):
        self.message = m
    def __str__(self):
        return self.message

class TokenError(Exception):
    def __init__(self, m):
        self.message = m
    def __str__(self):
        return self.message

class KeyError(Exception):
    '''Raised whenever a undefined key is used'''
    def __init__(self, m):
        self.message = m
    def __str__(self):
        return self.message

```

In [7]:

```

def UNameCheck(uname):#validation
    SpecialSym=['$', '@', '#', '%', '!']
    val = True

    if len(uname) <=0:
        raise ValueError('Value Too Small') #Value not defined
        val = False

    if len(uname) > 100:
        raise ValueError('Value Too Large')
        val = False

    if any(char.isdigit() for char in uname):
        raise ValueError('Not Numeral')
        val = False

    if any(char in SpecialSym for char in uname):
        raise ValueError('Special Symbol')
        val = False

    if val:
        return val

```

In [8]:

```

def PwdCheck(pwd): #validation

    SpecialSym=['$', '@', '#', '%', '!']

```

```

SpecialSym = [ '@', '!', '#', '$', '%', '&', '*', '^', '~', '._' ]
val = True

if len(pwd) < 6:
    raise ValueError('Value Too Small')
    val = False

if len(pwd) > 20:
    raise ValueError('Value Too Large')
    val = False

if not any(char.isdigit() for char in pwd):
    raise ValueError('No Numerals')
    val = False

if not any(char.isupper() for char in pwd):
    raise ValueError('Atleast an UpperCase Letter')
    val = False

if not any(char.islower() for char in pwd):
    raise ValueError('Atleast a LowerCase Letter')
    val = False

if not any(char in SpecialSym for char in pwd):
    raise ValueError('Special Symbol Not Allowed')
    val = False

if val:
    return val

```

In [9]:

```

def DbCheck(dno): #validation
    val = True
    if len(dno) < 6:
        raise ValueError('Value Too Small')
        val = False

    if len(dno) > 20:
        raise ValueError('Value Too Large')
        val = False

    if not any(char.isdigit() for char in dno):
        raise ValueError('No Numerals')
        val = False

    if not any(char.isupper() for char in dno):
        raise ValueError('Atleast an UpperCase Letter')
        val = False

    if val:
        return val

```

In [10]:

```

def PinCheck(pno): #validation
    val = True
    if len(pno) < 1:
        raise ValueError('Value Too Small')
        val = False

    if len(pno) > 5:
        raise ValueError('Value Too Large')
        val = False

    if not pno.isalpha():
        raise ValueError('Alphabets Not Allowed')
        val = False

    if val:
        return val

```

In [11]:

```
def IB():#internetbankingwelcomeloop
    print("-----")
    print("Hello User!\n\t\tWelcome to online banking facility.We, introduce a whole new set of f
unctions,for helping you out!Check out our features!\n1.Update Nominee\n2.Cash Withdrawal\n3.EMI C
alculator\n4.Checking Balance\n")
    print("-----")
    x=input("Choose")
    try:#try-exceptvalidation
        if x == '1':
            UN()
            cont()
        elif x== '2':
            CW()
            cont()
        elif x== '3':
            EMI()
            cont()
        elif x=='4':
            CB()
            cont()
    except ValueError:#ValueErrorbeingraisedwhenoutofvaluecomes
        raise ValueError
```

In [12]:

```
def ANO(ano):#accountnovalidation
    val = True
    if len(ano) < 1:
        raise ValueError('Value Too Small')
        val = False

    if len(ano) > 25:
        raise ValueError('Value Too Large')
        val = False

    if ano.isalpha():
        raise NonAlphabetError('Alphabets Not Allowed')
        val = False

    if val:
        return val
```

In [13]:

```
def NNO(nno):
    val = True
    if not any(char.isdigit() for char in pno):
        raise NotNumeralError('No Numerals')
        val = False

    if val:
        return val
```

In [14]:

```
def DOB(dob):
    val = True
    if dob.isalpha():
        raise NonAlphabetError('Alphabets Not Allowed')
        val = False

    if val:
        return val
```

In [15]:

```
def MNO(phno):
    val = True
    if len(phno) < 1:
        raise ValueError('Value Too Small')
        val = False
```

```

        val = raise

    if len(phno) > 25:
        raise ValueError('Value Too Large')
        val = False

    if phno.isalpha():
        raise NonAlphabetError('Alphabets Not Allowed')
        val = False

    if val:
        return val

```

In [16]:

```

def UN():#UpdateNominee
    datastore = { "token": 22331,"token": 22335}
    print("Dear Customers,\n\t\tWe are pleased to announce the Online Nomination facility, which i
s now available on NetBanking.All Resident Individuals who would have single holding Accounts and
Deposits with us, can now update or modify their nominee online.The modifications will reflect rea
l time.")
    print("\nAdding a nominee for all your Accounts and Deposits with AXIS Bank is now simple and
convenient.\nRegards,\nAXIS Bank")

    val = True
    def searchKeysByVal(dict, byVal):
        keysList = []
        itemsList = dict.items()
        for item in itemsList:
            if item[1] == byVal:
                keysList.append(item[0])
        return keysList

    x=int(input("Enter your Token-Id"))
    keysList = searchKeysByVal(datastore, x)

    for index, token in enumerate(keysList):
        val = True

    while True:
        #print("grf")

        listt=['dead','break-up','divorced']
        x=input("Enter your reason for updation")
        if x in listt:
            print("Your request for updation is accepted.")
            print("Enter nominee details")
            ano=input("Add account no")#dovalidation
            nno=input("Add name of nominee")
            dob=input("Add DOB")#dovalidation
            rshp=input("Enter rtlttnshp")
            add=input("Enter address")
            y=input("Are you sure?")
            if y=='y':
                print(tabulate([[ano, nno,dob,rshp,add,]], headers=['Account Number','Name Of the N
ominee', 'Date Of Birth','Relationship Status','Address'], tablefmt='orgtbl'))
                break
            else:
                raise TokenError

```

In [17]:

```

def CW():#CashWithdrawal
    master = Tk()
    def var_states():
        print("Credit Card: %d,\nDebit Card: %d" % (var1.get(), var2.get()))#toshowpop-up

    Label(master, text="Type of Card:").grid(row=0, sticky=W)
    var1 = IntVar()
    Checkbutton(master, text="credit", variable=var1).grid(row=1, sticky=W)
    var2 = IntVar()
    Checkbutton(master, text="debit", variable=var2).grid(row=2, sticky=W)
    Button(master, text='Quit', command=master.quit).grid(row=3, sticky=W, pady=4)
    Button(master, text='Show', command=var_states).grid(row=4, sticky=W, pady=4)

```

```

mainloop()

name=input("Enter name of card holder")#validation
UNameCheck(name)
dob=input("Enter DOB")#validation
DOB(dob)
cvv=input("Enter the CVV ")#validation
PinCheck(cvv)
phno=input("Enter the mobile number")#validation
MNO(phno)
amnt=input("Enter the amount")
print("-----")

print("-----")

x=input("Do you want to proceed?")
print("-----")

print("-----")

if x=='y' or x=='Y' or x=='yes' or x=='Yes':
    print("Your transaction is being processed")

else:
    raise KeyError

```

In [18]:

```

def EMI():#EMICalculatorvalidation
def emi(p, r, t):
    # for one month interest
    r = r/(12*100)
    # for one month period
    t = t*12
    emi = (p*r*pow(1+r,t))/(pow(1+r,t)-1)
    return emi

principal = int(input("Enter your Principal Amount"))
rate = int(input("rate of interest calculated on monthly basis "))
time = int(input("Loan Tenure"))

r = rate/(12*100)
t = time*12

print("Monthly EMI is= ", round((principal*r*pow(1+r,t))/(pow(1+r,t)-1)))

```

In [19]:

```

def CB():#checkbalancevalidation
data = {22331 : 5000, 22335:7896,22339 : 8000,22346:80000,223389:90000}
t = int(input("Enter the token"))
if t in data:
    print(f'{t},The remaining balance amount is {data[t]}')
else:
    raise TokenError('Token error')

```

In [20]:

```

def cont():
print()
print("Do you want to continue?(y/n)")
cont=input("Enter")

if cont=="y" or cont=="Y":
    IB()
else:
    print("Thank you")

```

In [21]:

```

#MAIN
print("-----")

```

```

-----")
print("\t\t\t\tWelcome to Internet Banking Support Of AXIS BANK\t\t\t\t")
print("-----")
print("Internet Banking is a convenient way to do banking from the comfort of your home or office.
Avoid the queue or delays and try our simple and secure Internet Banking facility for an unmatched
online banking experience.Just login today using your User ID and Password to experience AXIS Bank
Internet Banking.")
print("\nNOTE:")
print("Axis Bank does not send requests for Internet Banking Login ID, Password, Credit/ Debit car
d numbers, Bank account numbers or other sensitive financial information by email.")
print("Enter your Login Credentials:")
print("-----")
print("\t\t\t\tLogin either using your LoginId or Debit Card Number\t\t\t\t")
print("-----")
x=input("Enter your choice:")
if x=='LoginId':
    uname=input("Enter Username:")
    UNameCheck(uname)
    pwd=input("Enter Password:")
    PwdCheck(pwd)
    IB()
elif x=='Debit':
    dno=input("Enter Debit Card Number:")
    DbCheck(dno)
    pno=input("Enter PIN Number:")
    PinCheck(pno)
    IB()
else:
    raise TypeError

```

```

-----
|      Welcome to Internet Banking Support Of AXIS BANK      |
-----

Internet Banking is a convenient way to do banking from the comfort of your home or office. Avoid
the queue or delays and try our simple and secure Internet Banking facility for an unmatched onlin
e banking experience.Just login today using your User ID and Password to experience AXIS Bank
Internet Banking.

NOTE:
Axis Bank does not send requests for Internet Banking Login ID, Password, Credit/ Debit card numbe
rs, Bank account numbers or other sensitive financial information by email.
Enter your Login Credentials:
-----
|      Login either using your LoginId or Debit Card Number      |
-----

Enter your choice:LoginId
Enter Username:Riyaelii
Enter Password:Pulickans@22
-----

Hello User!
    Welcome to online banking facility.We,introduce a whole new set of functions,for helping you ou
t!Check out our features!
1.Update Nominee
2.Cash Withdrawal
3.EMI Calculator
4.Checking Balance
-----

Choose1
Dear Customers,
    We are pleased to announce the Online Nomination facility, which is now available on
NetBanking.All Resident Individuals who would have single holding Accounts and Deposits with us, c
an now update or modify their nominee online.The modifications will reflect real time.

Adding a nominee for all your Accounts and Deposits with AXIS Bank is now simple and convenient.
Regards,
AXIS Bank
Enter your Token-Id22331

```

Enter your reason for updationdead  
Your request for updation is accepted.  
Enter nominee details  
Add account noASDF45678  
Add name of nomineeVarkey  
Add DOB22/10/1997  
Enter rtlttnshpHusband  
Enter addressmkld  
Are you sure?y

| Account Number | Name Of the Nominee | Date Of Birth | Relationship Status | Address |
|----------------|---------------------|---------------|---------------------|---------|
| ASDF45678      | Varkey              | 22/10/1997    | Husband             | mkld    |

Do you want to continue?(y/n)  
Enter y

Hello User!

Welcome to online banking facility.We, introduce a whole new set of functions,for helping you out!Check out our features!

- 1.Update Nominee
- 2.Cash Withdrawal
- 3.EMI Calculator
- 4.Checking Balance

Choose2

Credit Card: 1,  
Debit Card: 0  
Enter name of card holderRIYAA  
Enter DOB18/08/1998  
Enter the CVV 7896  
Enter the mobile number3456789  
Enter the amount4888

Do you want to proceed?Y

Your transaction is being processed

Do you want to continue?(y/n)  
Enter Y

Hello User!

Welcome to online banking facility.We, introduce a whole new set of functions,for helping you out!Check out our features!

- 1.Update Nominee
- 2.Cash Withdrawal
- 3.EMI Calculator
- 4.Checking Balance

Choose3

Enter your Principal Amount4567893  
rate of interest calculated on monthly basis 9  
Loan Tenure6  
Monthly EMI is= 82339

Do you want to continue?(y/n)  
Enter Y

Hello User!

Welcome to online banking facility.We, introduce a whole new set of functions,for helping you out!Check out our features!

- 1.Update Nominee
- 2.Cash Withdrawal
- 3.EMI Calculator
4. Checking Balance



4.Checking Balance

Choose4

Enter the token22331

22331,The remaining balance amount is 5000

Do you want to continue?(y/n)

Enter y

Hello User!

Welcome to online banking facility.We, introduce a whole new set of functions,for helping you out!Check out our features!

- 1.Update Nominee
- 2.Cash Withdrawal
- 3.EMI Calculator
- 4.Checking Balance

Choose2

Credit Card: 1,

Debit Card: 0

Enter name of card holder85366

**NotNumeralError**

Traceback (most recent call last)

<ipython-input-21-f95f287f9b9c> in <module>

16 pwd=input("Enter Password:")

17 PwdCheck(pwd)

---> 18 IB()

19 elif x=='Debit':

20 dno=input("Enter Debit Card Number:")

<ipython-input-11-7dc9dd22080f> in IB()

7 if x == '1':

8 UN()

----> 9 cont()

10 elif x== '2':

11 CW()

<ipython-input-20-f469b8150390> in cont()

5

6 if cont=="y" or cont=="Y":

----> 7 IB()

8 else:

9 print("Thank you")

<ipython-input-11-7dc9dd22080f> in IB()

10 elif x== '2':

11 CW()

---> 12 cont()

13 elif x== '3':

14 EMI()

<ipython-input-20-f469b8150390> in cont()

5

6 if cont=="y" or cont=="Y":

----> 7 IB()

8 else:

9 print("Thank you")

<ipython-input-11-7dc9dd22080f> in IB()

13 elif x== '3':

14 EMI()

---> 15 cont()

16 elif x=='4':

17 CB()

<ipython-input-20-f469b8150390> in cont()

5

6 if cont=="y" or cont=="Y":

----> 7 IB()

8 else:

9 print("Thank you")

```
<ipython-input-11-7dc9dd22080f> in IB()
    16         elif x=='4':
    17             CB()
--> 18             cont()
    19     except ValueError:#ValueErrorbeingraisedwhenoutofvaluecomes
    20         raise ValueError
```

```
<ipython-input-20-f469b8150390> in cont()
     5
     6     if cont=="y" or cont=="Y":
--> 7         IB()
     8     else:
     9         print("Thank you")
```

```
<ipython-input-11-7dc9dd22080f> in IB()
     9         cont()
    10         elif x== '2':
--> 11             CW()
    12             cont()
    13             elif x== '3':
```

```
<ipython-input-17-dac0335a118b> in CW()
    14
    15     name=input("Enter name of card holder")#validation
--> 16     UNameCheck(name)
    17     dob=input("Enter DOB")#validation
    18     DOB(dob)
```

```
<ipython-input-7-ee3a68e12586> in UNameCheck(uname)
    12
    13     if any(char.isdigit() for char in uname):
--> 14         raise NotNumeralError('Not Numeral')
    15         val = False
    16
```

**NotNumeralError:** Not Numeral

In [ ]:

In [ ]:

In [ ]: