1,

class customer:

def \_\_init\_\_(self,name,ph\_no):

self.name=name

self.ph\_no=ph\_no

def cus\_display(self):

print(f"NAME:{self.name}\nPHONE\_NO:{self.ph\_no}")

class depositor(customer):

def \_\_init\_\_(self,name,ph\_no,acc\_no,balance):

super().\_\_init\_\_(name,ph\_no)

self.acc\_no=acc\_no

self.balance=balance

def dep\_display(self):

self.cus\_display()

print(f"ACCOUNT\_NO:{self.acc\_no}\nBALANCE:{self.balance}")

class borrower(depositor):

def \_\_init\_\_(self,name,ph\_no,acc\_no,balance,loan\_no,loan\_amount):

super().\_\_init\_\_(name,ph\_no,acc\_no,balance)

self.loan\_no=loan\_no

self.loan\_amount=loan\_amount

def b\_display(self):

self.dep\_display()

print(f"LOAN\_NO:{self.loan\_no}\nLOAN\_AMOUNT:{loan\_amount}")

l=[]

p=int(input("enter the number of customer details to be added:"))

for \_ in range(p):

name=input("name:")

ph\_no=int(input("enter the phone number:"))

acc\_no=int(input("enter the account number:"))

balance=float(input("enter the account balance:"))

loan\_no=int(input("enter the loan number:"))

loan\_amount=input("enter the loan amount:")

l.append(borrower(name,ph\_no,acc\_no,balance,aloan\_no,loan\_amount))

for customer in l:

customer.b\_display()

2,

def findMaxLength(nums):

count\_map={0:-1}

max\_length=0

count=0

for i, num in enumerate(nums):

count+=1 if num==1 else -1

if count in count\_map:

max\_length=max(max\_length,i-count\_map[count])

else:

count\_map[count]=i

return max\_length

nums1=[0,1,0,1,1,0,1,0]

print(findMaxLength(nums1))