Name: Abdulmuiz Khalid Shaikh
Roll no.:2101062
Subject: Data Structure and Algorithm Laboratory
Assignment No.1

```
 --------------------------------------------------------------------------
Module Code:
--------------------------------------------------------------------------
class Record:
def __init__(self):
self.name=None
self.number=None
def get_name(self):
return self.name
def get_number(self):
return self.number
def set_number(self,numb): #numb=N
self.number=numb
def set_name(self,name):
self.name=name
def __str__(self):
record="Name: "+str(self.get_name())+"\tNumber: "+str(self.get_number())
return record
class Hashtable():
def __init__(self):
self.size=int(input("Enter the size of Hash Table "))
self.table= list(None for i in range(self.size))
self.elementcount=0
self.comparision=0
def isFull(self):
if self.size==self.elementcount:
return True
```

```
else:
return False
def hashfun(self,element):
return element%self.size
def insert(self,record):
if self.isFull():
print(" \nHash Table is Full ")
else:
position=self.hashfun(record.get_number())
if self.table[position]==None:
self.table[position]=record
print("Phone number of "+str(record.get_name()).lower()+" is at position "+str(position))
self.elementcount +=1
else:
print("\nCollision has occured for "+str(record.get_name()).lower()+"'s phone number at position
"+str(position)+"finding new position ")
while self.table[position]!=None:
position +=1
if position >=self.size:
position=0
self.table[position]=record
print("phone number of"+record.get_name()+"is at position"+str(position))
self.elementcount +=1
def display(self):
print("\n")
```

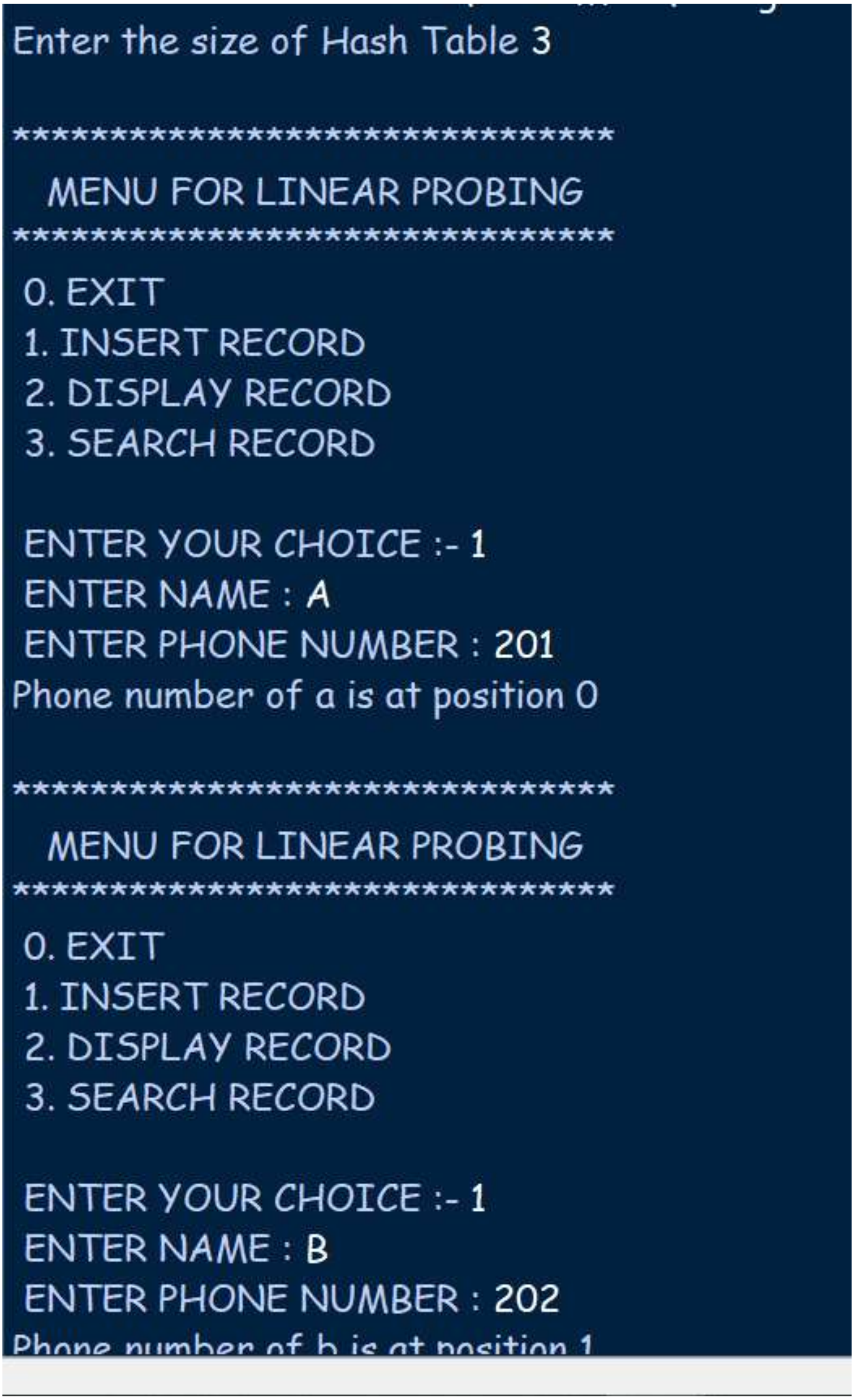
```
for i in range(self.size):
if(self.table[i]!=None):
print("Hash value "+str(i)+"\t"+str(self.table[i]))
print("\nThe total record in table are "+str(self.elementcount))
def search(self,record):
position=self.hashfun(record.get_number())
is_found=False
while(self.comparision<self.size):
if(self.table[position]==None):
self.comparision+=1
return -1
else:
if(str(self.table[position].get_name()).lower()==str(record.get_name()).lower() and
self.table[position].get_number()==record.get_number()):
self.comparision+=1
return position
position+=1
self.comparision += 1
if(position>=self.size):
position=0
if(is_found==False):
return -1
def input_record():
recod=Record()
```

```
name=input(" ENTER NAME : ")
N=int(input(" ENTER PHONE NUMBER : "))
recod.set_name(name)
recod.set_number(N)
return recod
def main():
h1 = Hashtable()
Menu='MENU FOR LINEAR PROBING'
check=True
while(check==True):
print("\n******************************")
print(Menu.center(30,))
print("******************************")
print(" 0. EXIT")
print(" 1. INSERT RECORD")
print(" 2. DISPLAY RECORD")
print(" 3. SEARCH RECORD")
choice = int(input("\n ENTER YOUR CHOICE :- "))
if(choice==1):
record=input_record()
h1.insert(record)
elif(choice==2):
h1.display()
elif(choice==3):
print("\n ENTER THE RECORD YOU WANT TO SEARCH :- ")
r=input_record()
value=h1.search(r)
if(value==-1):
print("RECORD NOT FOUND ")
```

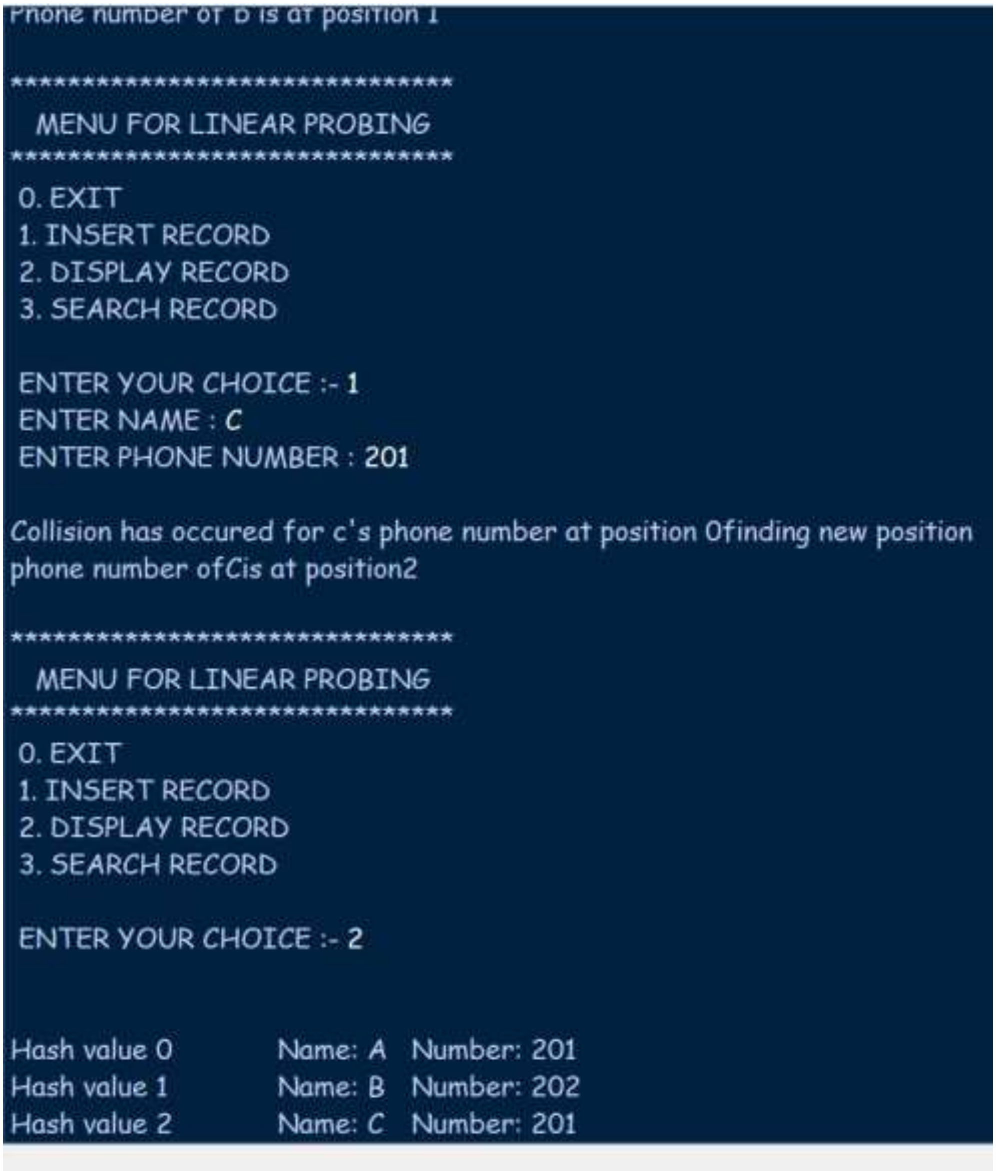
```
else:
print("RECORD IS AT POSITION :- "+str(value))
elif(choice==0):
check=False
else:
print("\nWRONG CHOICE")
main()
```

Main driver file code:

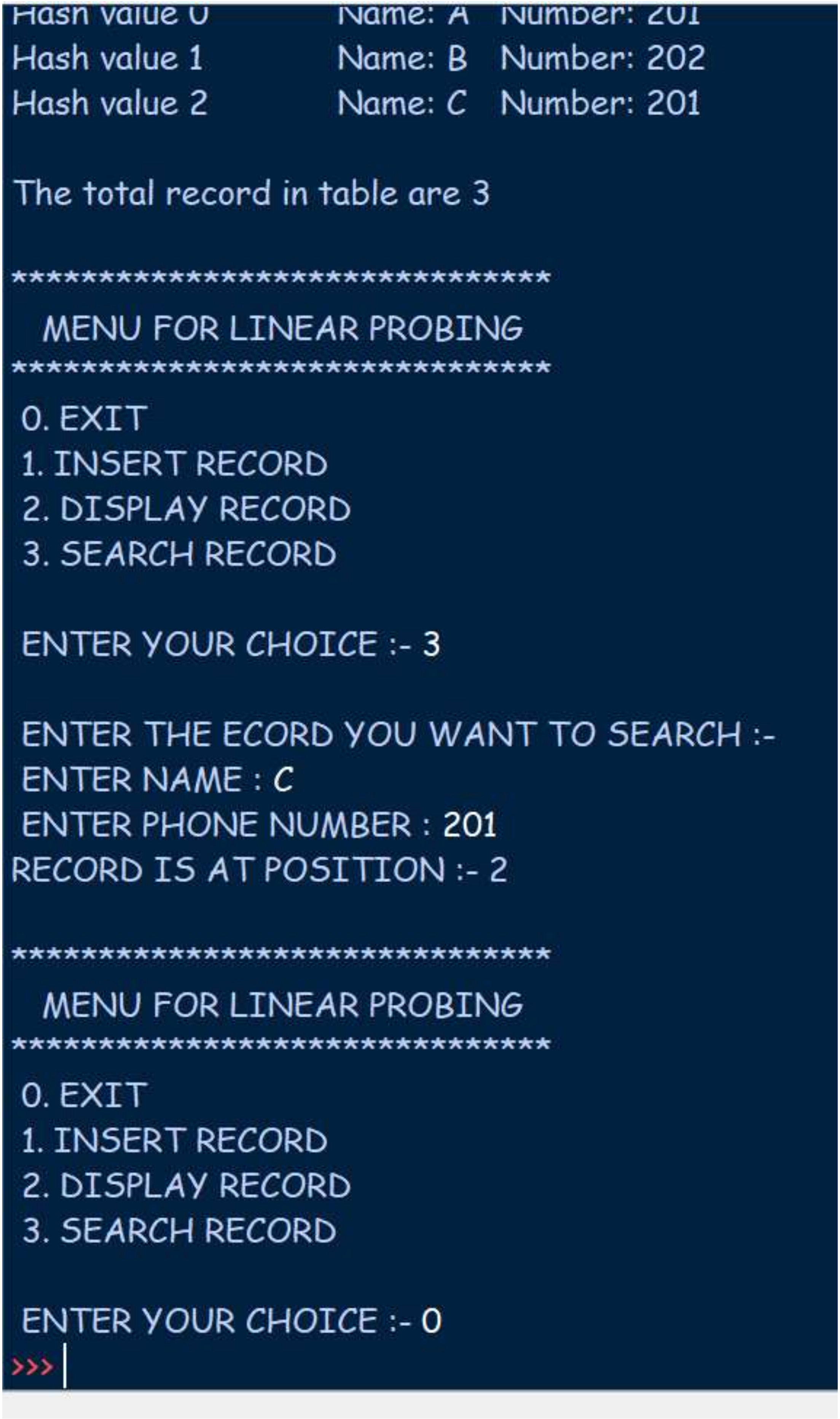
```
def linear():
import main
def double_hasing():
import main_2
def main():
checker=True
Menu='MAIN-MENU'
while(checker):
print("\n############################")
print(Menu.center(30,))
print("############################")
print(" 0. EXIT")
print(" 1. Linear Probing")
print(" 2. Double Hashing")
op = int(input("\n ENTER YOUR CHOICE :- "))
if(op==0):
checker=False
elif(op==1):
linear()
elif(op==2):
double_hasing()
else:
print("\nWRONG CHOICE\n")
main()
------------------------------------------------------------------------------
------------------------------------------------------------------------------
```



```
Enter the size of Hash Table 3

********************************
  MENU FOR LINEAR PROBING
********************************
 0. EXIT
 1. INSERT RECORD
 2. DISPLAY RECORD
 3. SEARCH RECORD

 ENTER YOUR CHOICE :- 1
 ENTER NAME : A
 ENTER PHONE NUMBER : 201
Phone number of a is at position 0

********************************
  MENU FOR LINEAR PROBING
********************************
 0. EXIT
 1. INSERT RECORD
 2. DISPLAY RECORD
 3. SEARCH RECORD

 ENTER YOUR CHOICE :- 1
 ENTER NAME : B
 ENTER PHONE NUMBER : 202
Phone number of b is at position 1
```



```
Phone number of B is at position 1

********************************
  MENU FOR LINEAR PROBING
********************************
 0. EXIT
 1. INSERT RECORD
 2. DISPLAY RECORD
 3. SEARCH RECORD

 ENTER YOUR CHOICE :- 1
 ENTER NAME : C
 ENTER PHONE NUMBER : 201

Collision has occured for c's phone number at position 0finding new position
phone number ofCis at position2

********************************
  MENU FOR LINEAR PROBING
********************************
 0. EXIT
 1. INSERT RECORD
 2. DISPLAY RECORD
 3. SEARCH RECORD

 ENTER YOUR CHOICE :- 2


Hash value 0        Name: A   Number: 201
Hash value 1        Name: B   Number: 202
Hash value 2        Name: C   Number: 201
```



```
Hash value 0          Name: A   Number: 201
Hash value 1          Name: B   Number: 202
Hash value 2          Name: C   Number: 201

The total record in table are 3

********************************
  MENU FOR LINEAR PROBING
********************************
 0. EXIT
 1. INSERT RECORD
 2. DISPLAY RECORD
 3. SEARCH RECORD

 ENTER YOUR CHOICE :- 3

 ENTER THE ECORD YOU WANT TO SEARCH :-
 ENTER NAME : C
 ENTER PHONE NUMBER : 201
RECORD IS AT POSITION :- 2

********************************
  MENU FOR LINEAR PROBING
********************************
 0. EXIT
 1. INSERT RECORD
 2. DISPLAY RECORD
 3. SEARCH RECORD

 ENTER YOUR CHOICE :- 0
>>>
```