PRACTICAL NO: 1

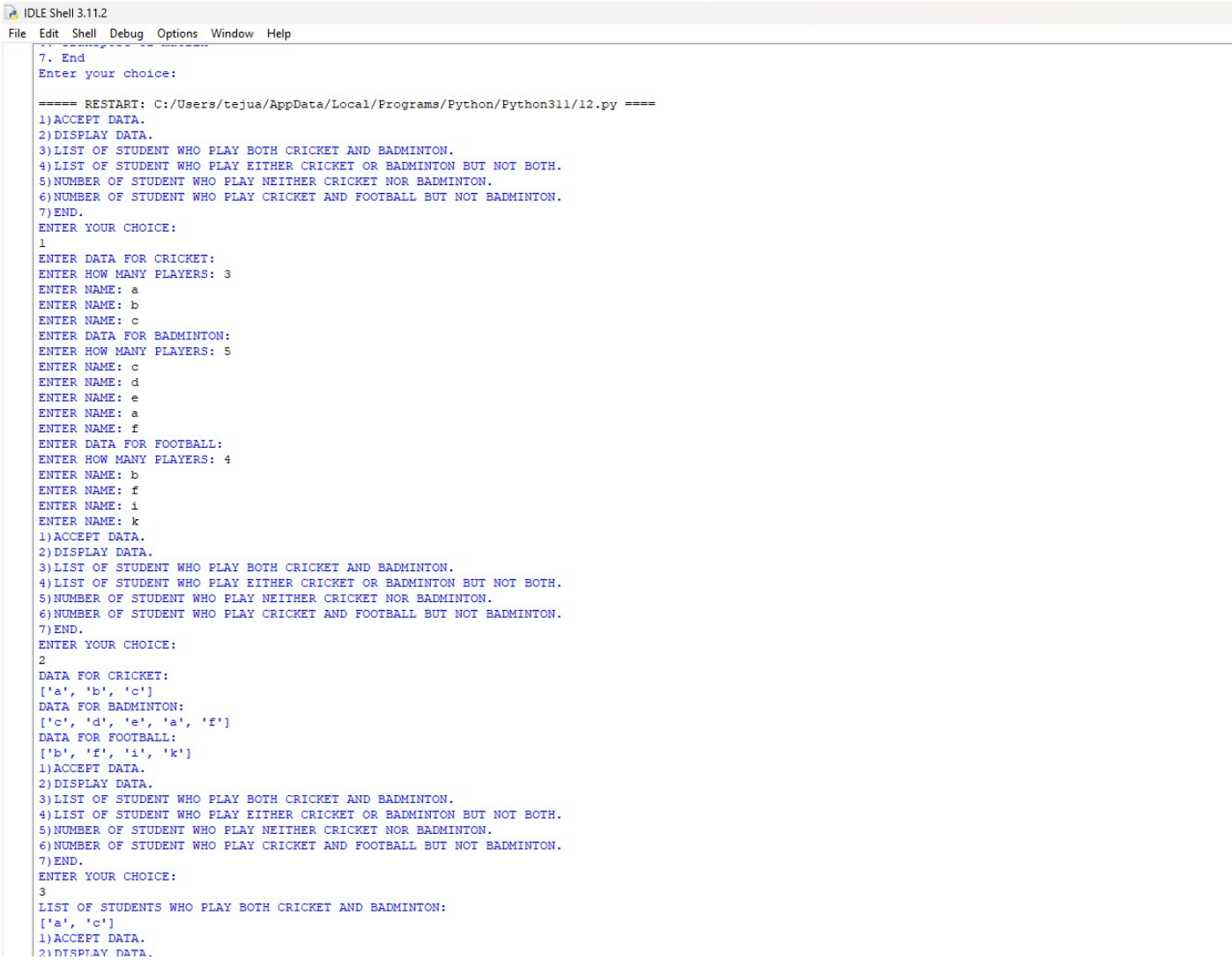
'''Group A1 in second year computer engg class, group A students play cricket, group B students play badminton and group C students play football. Write a python program using functions to compute the following:
1)List of students who play both cricket and badminton.
2)List of students who play either cricket or badminton but not both.
3)Number of student who play neither cricket nor badminton.
4)Number of student who play cricket and football but not badminton.
NOTE:While realizing the group, duplicate entry should be avoided, do not use set built in function.'''

```
def main():
   cricket=[]
   badminton=[]
   football=[]
   while True:
      print("1)ACCEPT DATA.")
      print("2)DISPLAY DATA.")
      print("3)LIST OF STUDENT WHO PLAY BOTH CRICKET AND BADMINTON.")
      print("4)LIST OF STUDENT WHO PLAY EITHER CRICKET OR BADMINTON BUT NOT
BOTH.")
      print("5)NUMBER OF STUDENT WHO PLAY NEITHER CRICKET NOR
BADMINTON.")
      print("6)NUMBER OF STUDENT WHO PLAY CRICKET AND FOOTBALL BUT NOT
BADMINTON.")
      print("7)END.")
      print("ENTER YOUR CHOICE:")
      choice=int(input())
      if(choice==1):
         print("ENTER DATA FOR CRICKET:")
         Accept(cricket)
         print("ENTER DATA FOR BADMINTON:")
         Accept(badminton)
         print("ENTER DATA FOR FOOTBALL:")
         Accept(football)
      elif(choice==2):
         print("DATA FOR CRICKET:")
         print(cricket)
         print("DATA FOR BADMINTON:")
         print(badminton)
         print("DATA FOR FOOTBALL:")
         print(football)
      elif(choice==3):
         cricket_badminton(cricket,badminton)
      elif(choice==4):
         either_cricket_badminton(cricket,badminton)
```

```
        elif(choice==5):
            neither_cricket_nor_badminton(cricket,badminton,football)
        elif(choice==6):
            either_cricket_football_nor_badminton(cricket,badminton,football)
        elif(choice==7):
            break;

def Accept(player):
    n=int(input("ENTER HOW MANY PLAYERS: "))
    for i in range(n):
        x=input("ENTER NAME: ")
        player.append(x)

def cricket_badminton(cricket,badminton):
    cr_bad=[]
    for i in range(len(cricket)):
        x=cricket[i]
        if x in badminton:
            cr_bad.append(x)
    print("LIST OF STUDENTS WHO PLAY BOTH CRICKET AND BADMINTON: ")
    print(cr_bad)

def either_cricket_badminton(cricket,badminton):
    final=[]
    for a in range(len(cricket)):
        final.append(cricket[a])
    output=[]
    for i in range(len(badminton)):
        if(badminton[i] not in cricket):
            final.append(badminton[i])
    Intersection=[]
    for i in range(len(cricket)):
        if(cricket[i] in badminton):
            Intersection.append(cricket[i])
    for i in range(len(final)):
        if(final[i] not in Intersection):
            output.append(final[i])
    print("LIST OF STUDENTS WHO PLAY EITHER CRICKET OR BADMINTON BUT NOT
BOTH: ")
    print(output)

def neither_cricket_nor_badminton(cricket,badminton,football):
    final=[]
    for i in range(len(cricket)):
        final.append(cricket[i])
    for i in range(len(badminton)):
        if(badminton[i] not in final):
            final.append(badminton[i])
    output=[]
```

```
    for i in range(len(football)):
        if(football[i] not in final):
            output.append(football[i])
    print("THE STUDENT NEITHER PLAY CRICKET NOR BADMINTON ARE: ")
    print(output)

def either_cricket_football_nor_badminton(cricket,badminton,football):
    final=[]
    output=[]
    for i in range(len(cricket)):
        if (cricket[i] in football):
            final.append(cricket[i])
    for i in range(len(final)):
        output.append(final[i])
    print("THE STUDENT WHO PLAY CRICKET AND FOOTBALL BUT NOT BADMINTON: ")
    print(output)

main()
```

O/P:



IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
DATA FOR CRICKET:
['a', 'b', 'c']
DATA FOR BADMINTON:
['c', 'd', 'e', 'a', 'f']
DATA FOR FOOTBALL:
['b', 'f', 'i', 'k']
1)ACCEPT DATA.
2)DISPLAY DATA.
3)LIST OF STUDENT WHO PLAY BOTH CRICKET AND BADMINTON.
4)LIST OF STUDENT WHO PLAY EITHER CRICKET OR BADMINTON BUT NOT BOTH.
5)NUMBER OF STUDENT WHO PLAY NEITHER CRICKET NOR BADMINTON.
6)NUMBER OF STUDENT WHO PLAY CRICKET AND FOOTBALL BUT NOT BADMINTON.
7)END.
ENTER YOUR CHOICE:
3
LIST OF STUDENTS WHO PLAY BOTH CRICKET AND BADMINTON:
['a', 'c']
1)ACCEPT DATA.
2)DISPLAY DATA.
3)LIST OF STUDENT WHO PLAY BOTH CRICKET AND BADMINTON.
4)LIST OF STUDENT WHO PLAY EITHER CRICKET OR BADMINTON BUT NOT BOTH.
5)NUMBER OF STUDENT WHO PLAY NEITHER CRICKET NOR BADMINTON.
6)NUMBER OF STUDENT WHO PLAY CRICKET AND FOOTBALL BUT NOT BADMINTON.
7)END.
ENTER YOUR CHOICE:
4
LIST OF STUDENTS WHO PLAY EITHER CRICKET OR BADMINTON BUT NOT BOTH:
['b', 'd', 'e', 'f']
1)ACCEPT DATA.
2)DISPLAY DATA.
3)LIST OF STUDENT WHO PLAY BOTH CRICKET AND BADMINTON.
4)LIST OF STUDENT WHO PLAY EITHER CRICKET OR BADMINTON BUT NOT BOTH.
5)NUMBER OF STUDENT WHO PLAY NEITHER CRICKET NOR BADMINTON.
6)NUMBER OF STUDENT WHO PLAY CRICKET AND FOOTBALL BUT NOT BADMINTON.
7)END.
ENTER YOUR CHOICE:
5
THE STUDENT NEITHER PLAY CRICKET NOR BADMINTON ARE:
['i', 'k']
1)ACCEPT DATA.
2)DISPLAY DATA.
3)LIST OF STUDENT WHO PLAY BOTH CRICKET AND BADMINTON.
4)LIST OF STUDENT WHO PLAY EITHER CRICKET OR BADMINTON BUT NOT BOTH.
5)NUMBER OF STUDENT WHO PLAY NEITHER CRICKET NOR BADMINTON.
6)NUMBER OF STUDENT WHO PLAY CRICKET AND FOOTBALL BUT NOT BADMINTON.
7)END.
ENTER YOUR CHOICE:
6
THE STUDENT WHO PLAY CRICKET AND FOOTBALL BUT NOT BADMINTON:
['b']
1)ACCEPT DATA.
2)DISPLAY DATA.
3)LIST OF STUDENT WHO PLAY BOTH CRICKET AND BADMINTON.
4)LIST OF STUDENT WHO PLAY EITHER CRICKET OR BADMINTON BUT NOT BOTH.
5)NUMBER OF STUDENT WHO PLAY NEITHER CRICKET NOR BADMINTON.
6)NUMBER OF STUDENT WHO PLAY CRICKET AND FOOTBALL BUT NOT BADMINTON.
7)END.
ENTER YOUR CHOICE:
7
```

PRACTICAL NO:2

"'Write a python program to store marks scored in subject fundamental of data structure by N students in class.Write function to compute following:
1)The average score of class.
2)Highest score and lowest score of class.
3)Count of student who were absent for the test.
4)Display marks with highest frequency"'

```
def main():
    fds=[]
    while True:
        print("1)Enter Marks")
        print("2)Display Marks")
        print("3)Average of Marks")
        print("4)Highest score and lowest Score")
        print("5)Absent student Count")
        print("6)Display marks with Highest Frequency")
        print("7)End")
        print("Enter your choice: ")
        choice=int(input())
        if (choice==7):
            break
        elif(choice==1):
            accept_marks(fds)
        elif(choice==2):
            print("FDS Marks are: ")
            print(fds)
        elif(choice==3):
            average_marks(fds)
        elif(choice==4):
            highest_lowest_marks(fds)
        elif(choice==5):
            absent_count(fds)
        elif(choice==6):
            display_marks_with_highest_frequency(fds)
def accept_marks(fds):
    print("Enter number of students: ")
    n=int(input())
    for i in range(n):
        print("Enter marks: ")
        x=int(input())
        fds.append(x)
def average_marks(fds):
    total=0
    present=0
    for i in fds:
        if(i!=-1):
            total=total+i
```

```
            present=present+1
    average_marks=total/present
    print("Average is: ",average_marks)
def highest_lowest_marks(fds):
    highest=-1
    lowest=999
    for i in fds:
        if(i!=-1):
            if(i>highest):
                highest=i
            if(i<lowest):
                lowest=i
    print("Highest Marks is: ",highest)
    print("Lowest Marks is: ",lowest)
def absent_count(fds):
    count=0
    for i in fds:
        if(i==-1):
            count=count+1
    print("Absent student: ")
    print(count)
def display_marks_with_highest_frequency(fds):
    freq=0
    for i in range(len(fds)):
        count=0
        if(i!=-1):
            for j in range(len(fds)):
                if (fds[i]==fds[j]):
                    count=count+1
            if(freq<count):
                freq=count
                high_freq_mark=fds[i]
    print("Highest frequency marks is ",high_freq_mark)
    print("Highest frequency count is ",freq)
main()
```

O/P:


IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: E:/Data Structure/123.py ==========
1)Enter Marks
2)Display Marks
3)Average of Marks
4)Highest score and lowest Score
5)Absent student Count
6)Display marks with Highest Frequency
7)End
Enter your choice:
1
Enter number of students:
7
Enter marks:
90
Enter marks:
67
Enter marks:
78
Enter marks:
-1
Enter marks:
90
Enter marks:
-1
Enter marks:
66
1)Enter Marks
2)Display Marks
3)Average of Marks
4)Highest score and lowest Score
5)Absent student Count
6)Display marks with Highest Frequency
7)End
Enter your choice:
2
FDS Marks are:
[90, 67, 78, -1, 90, -1, 66]
1)Enter Marks
2)Display Marks
3)Average of Marks
4)Highest score and lowest Score
5)Absent student Count
6)Display marks with Highest Frequency
7)End
Enter your choice:
3
Average is: 78.2
1)Enter Marks
2)Display Marks
3)Average of Marks
4)Highest score and lowest Score
5)Absent student Count
6)Display marks with Highest Frequency
7)End
Enter your choice:
4

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
5)Absent student Count
6)Display marks with Highest Frequency
7)End
Enter your choice:
2
FDS Marks are:
[90, 67, 78, -1, 90, -1, 66]
1)Enter Marks
2)Display Marks
3)Average of Marks
4)Highest score and lowest Score
5)Absent student Count
6)Display marks with Highest Frequency
7)End
Enter your choice:
3
Average is:  78.2
1)Enter Marks
2)Display Marks
3)Average of Marks
4)Highest score and lowest Score
5)Absent student Count
6)Display marks with Highest Frequency
7)End
Enter your choice:
4
Highest Marks is:  90
Lowest Marks is:  66
1)Enter Marks
2)Display Marks
3)Average of Marks
4)Highest score and lowest Score
5)Absent student Count
6)Display marks with Highest Frequency
7)End
Enter your choice:
5
Absent student:
2
1)Enter Marks
2)Display Marks
3)Average of Marks
4)Highest score and lowest Score
5)Absent student Count
6)Display marks with Highest Frequency
7)End
Enter your choice:
6
Highest frequency marks is  90
Highest frequency count is  2
1)Enter Marks
2)Display Marks
3)Average of Marks
4)Highest score and lowest Score
5)Absent student Count
6)Display marks with Highest Frequency
7)End
Enter your choice:
7
```

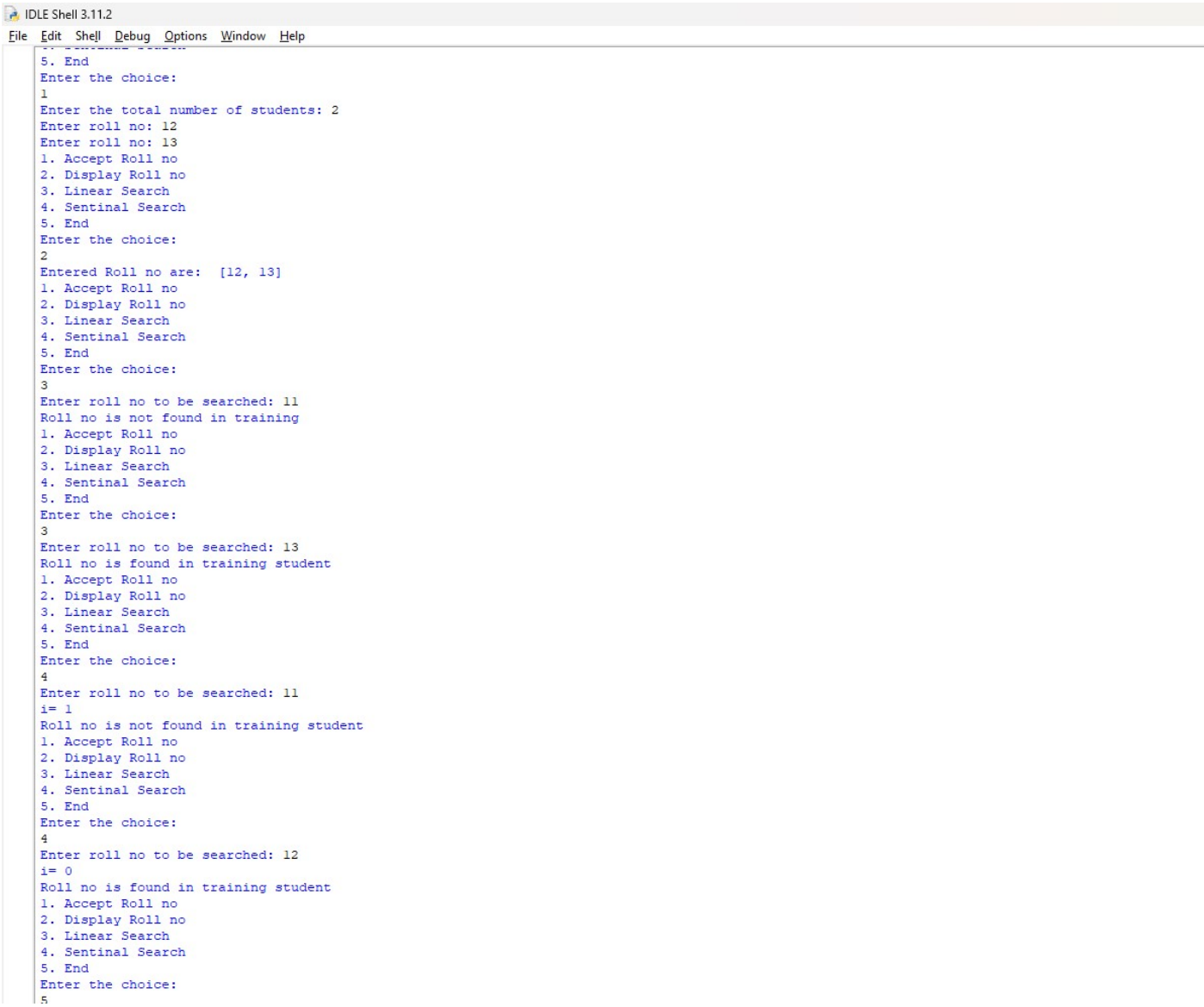
PRACTICAL NO:4

A)"'Write a python program to store roll no of students in array who attended training program in random order. Write function for searching whether particular student attending training program or not using linear search and sentinel search"'

```
def main():
    Roll_no=[]
    while True:
        print("1. Accept Roll no")
        print("2. Display Roll no")
        print("3. Linear Search")
        print("4. Sentinal Search")
        print("5. End")
        print("Enter the choice: ")
        choice=int(input())
        if(choice==1):
            Accept_Rollno(Roll_no)
        elif(choice==2):
            print("Entered Roll no are: ",Roll_no)
        elif(choice==3):
            Linear_search(Roll_no)
        elif(choice==4):
            Sentinal_search(Roll_no)
        elif(choice==5):
            break
def  Accept_Rollno(Roll_no):
    n=int(input("Enter the total number of students: "))
    for i in range(n):
        x=int(input("Enter roll no: "))
        Roll_no.append(x)

def Linear_search(Roll_no):
    n=len(Roll_no)
    search_key=int(input("Enter roll no to be searched: "))
    flag=0
    for i in range(n):
        if(search_key==Roll_no[i]):
            print("Roll no is found in training student")
            flag=1
            break
    if(flag==0):
        print("Roll no is not found in training")

def Sentinal_search(Roll_no):
    n=len(Roll_no)
    search_key=int(input("Enter roll no to be searched: "))
    last=Roll_no[n-1]
    Roll_no[n-1]=search_key
```

```
    i=0
    while(Roll_no[i]!=search_key):
        i=i+1
    print("i=",i)
    Roll_no[n-1]=last
    if((search_key==Roll_no[n-1])or(i<n-1)):
        print("Roll no is found in training student")
    else:
        print("Roll no is not found in training student")
main()
```

O/P:

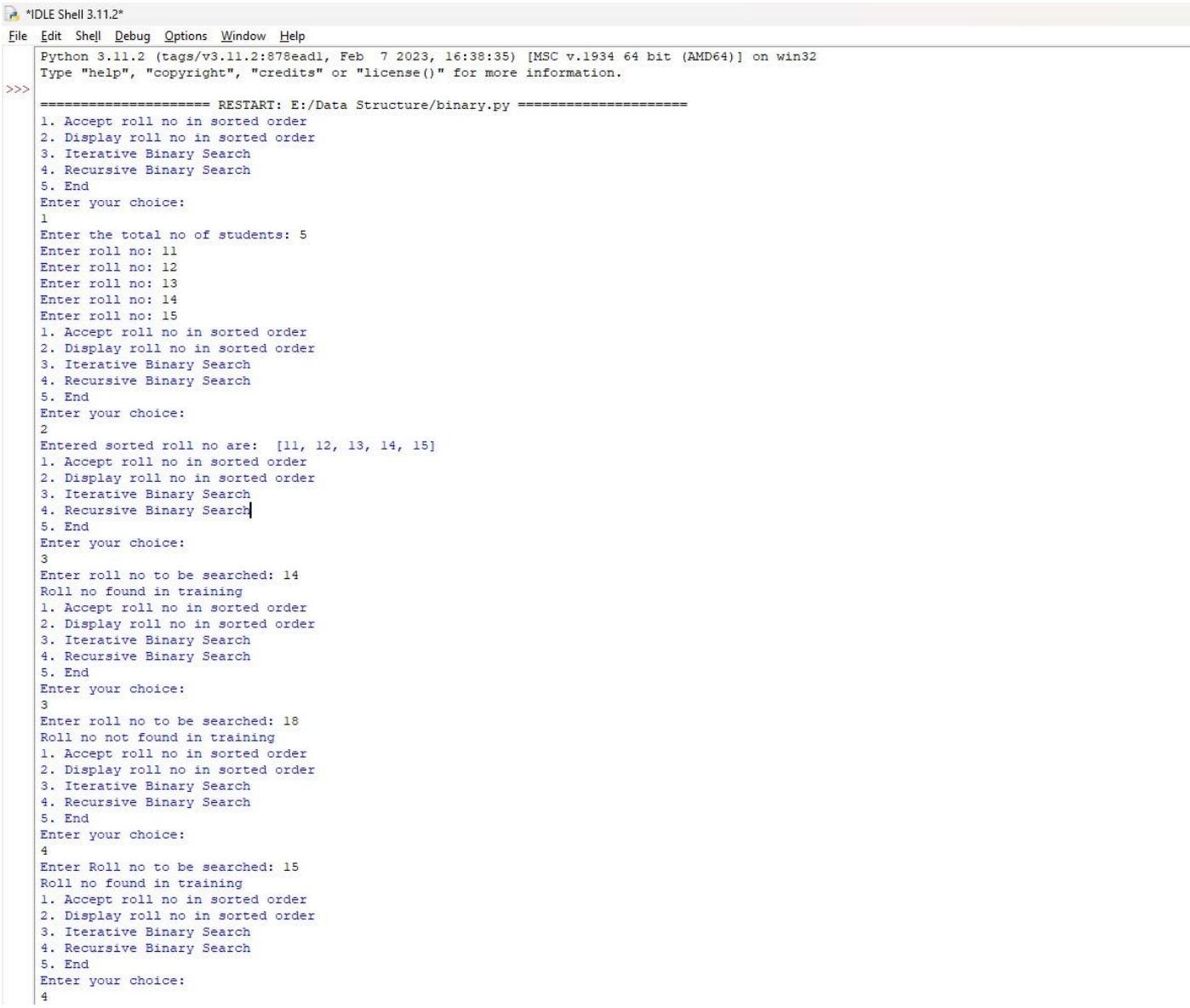

IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
5. End
Enter the choice:
1
Enter the total number of students: 2
Enter roll no: 12
Enter roll no: 13
1. Accept Roll no
2. Display Roll no
3. Linear Search
4. Sentinal Search
5. End
Enter the choice:
2
Entered Roll no are: [12, 13]
1. Accept Roll no
2. Display Roll no
3. Linear Search
4. Sentinal Search
5. End
Enter the choice:
3
Enter roll no to be searched: 11
Roll no is not found in training
1. Accept Roll no
2. Display Roll no
3. Linear Search
4. Sentinal Search
5. End
Enter the choice:
3
Enter roll no to be searched: 13
Roll no is found in training student
1. Accept Roll no
2. Display Roll no
3. Linear Search
4. Sentinal Search
5. End
Enter the choice:
4
Enter roll no to be searched: 11
i= 1
Roll no is not found in training student
1. Accept Roll no
2. Display Roll no
3. Linear Search
4. Sentinal Search
5. End
Enter the choice:
4
Enter roll no to be searched: 12
i= 0
Roll no is found in training student
1. Accept Roll no
2. Display Roll no
3. Linear Search
4. Sentinal Search
5. End
Enter the choice:
5

B)"'Write a python program to store roll no of students in array who attended training program in sorted order. Write function for searching whether particular student attending training program or not using binary search"'

```
def main():
    Roll_no=[]
    while True:
        print("1. Accept roll no in sorted order")
        print("2. Display roll no in sorted order")
        print("3. Iterative Binary Search")
        print("4. Recursive Binary Search")
        print("5. End")
        print("Enter your choice: ")
        choice=int(input())
        if(choice==5):
            break
        elif(choice==1):
            Accept_Rollno(Roll_no)
        elif(choice==2):
            print("Entered sorted roll no are: ",Roll_no)
        elif(choice==3):
            Iterative_binary_search(Roll_no)
        elif(choice==4):
            low=0
            high=len(Roll_no)-1
            key=int(input("Enter Roll no to be searched: "))
            Recursive_binary_search(Roll_no,low,high,key)

def Accept_Rollno(Roll_no):
    n=int(input("Enter the total no of students: "))
    for i in range(n):
        x=int(input("Enter roll no: "))
        Roll_no.append(x)

def Iterative_binary_search(Roll_no):
    low=0
    high=len(Roll_no)-1
    key=int(input("Enter roll no to be searched: "))
    while(low<=high):
        mid=int((low+high)/2)
        if (Roll_no[mid]==key):
            print("Roll no found in training")
            return
        elif(Roll_no[mid]<key):
            low=mid+1
        else:
            low=mid-1
    if(low>high):
        print("Roll no not found in training")
```

```
def Recursive_binary_search(Roll_no,low,high,key):
    if(low<=high):
        mid=int((low+high)/2)
        if (Roll_no[mid]==key):
            print("Roll no found in training")
            return
        elif(Roll_no[mid]<key):
            Recursive_binary_search(Roll_no,mid+1,high,key)
        else:
            Recursive_binary_search(Roll_no,low,mid-1,key)
    if(low>high):
        print("Roll no not found in training")

main()
```

O/P:




IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
5. End
Enter your choice:
1
Enter the total no of students: 5
Enter roll no: 11
Enter roll no: 12
Enter roll no: 13
Enter roll no: 14
Enter roll no: 15
1. Accept roll no in sorted order
2. Display roll no in sorted order
3. Iterative Binary Search
4. Recursive Binary Search
5. End
Enter your choice:
2
Entered sorted roll no are: [11, 12, 13, 14, 15]
1. Accept roll no in sorted order
2. Display roll no in sorted order
3. Iterative Binary Search
4. Recursive Binary Search
5. End
Enter your choice:
3
Enter roll no to be searched: 14
Roll no found in training
1. Accept roll no in sorted order
2. Display roll no in sorted order
3. Iterative Binary Search
4. Recursive Binary Search
5. End
Enter your choice:
3
Enter roll no to be searched: 18
Roll no not found in training
1. Accept roll no in sorted order
2. Display roll no in sorted order
3. Iterative Binary Search
4. Recursive Binary Search
5. End
Enter your choice:
4
Enter Roll no to be searched: 15
Roll no found in training
1. Accept roll no in sorted order
2. Display roll no in sorted order
3. Iterative Binary Search
4. Recursive Binary Search
5. End
Enter your choice:
4
Enter Roll no to be searched: 20
Roll no not found in training
1. Accept roll no in sorted order
2. Display roll no in sorted order
3. Iterative Binary Search
4. Recursive Binary Search
5. End
Enter your choice: