Assignment – 5
(List)

1. **Write a Python program to remove duplicates from a list**

Ans:-

```
l1=list(map(int,input().split()))

l2=list(set(l1))

print(l2)
```

Output:-

```
10 20 30 15 10 20 45 12

[10, 12, 45, 15, 20, 30]
```

2. **Write a Python program to find the second largest number in a list.**

Answer:-

```
l1=list(map(int,input().split()))

l2=list(set(l1))

l2.sort(reverse=True)

print(l2[1])
```

Output:-

```
12 3 14 26 37 12 24

26
```

3. **Write a Python program to sort a list of elements using the bubble sort algorithm.**

Answer:-

```
l1=list(map(int,input().split()))

for i in range(len(l1)-1):

for j in range(len(l1)-i-1):

if l1[j]>l1[j+1]:

temp=l1[j]

l1[j]=l1[j+1]

l1[j+1]=temp

print(l1)
```

Output :-

```
12 13 14 26 37 12 24
```

4. **Write a Python program to find the common elements between two lists.**
   Answer :-

```
l1=list(map(int,input().split()))
l2=list(map(int,input().split()))
for i in l1:
for j in l2:
if i==j:
print(i)
```

Output :

```
12 13 14 26 37 12 24
12 13 14 10 9 8
12
13
14
12
```

5. **Write a Python program to count the frequency of each element in a list.**
   Answer:-

```
l1=[10,23,10,25,37,45,17,45]
unique=list(set(l1))
for i in unique:
print(i," occurs ",l1.count(i),"times")
```

Output :-

```
25 occurs 1 times
 37 occurs 1 times
10 occurs 2 times
45 occurs 2 times
17 occurs 1 times
23 occurs 1 times
18
```

6. **Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings**
   Answer:-

```
lst=["anurag","roshan","ghosh","senapati","abhaya"]
count=0
for i in lst:
if len(i)>=2 and i[0]==i[-1]:
count+=1
print(count)
```

Output:-

```
1
```

7. **Write a Program that get two lists as input and check if they have at least one common member .**

Answer:-

```
l1=list(map(int,input().split()))
l2=list(map(int,input().split()))
for i in l1:
for j in l2:
if i==j:
common=1
if common==1:
print(True)
else :
print(False)
```

Output:-

```
  12 3 4 5 67 2 4 6 8 5
True
```

8. **Write a Python program to print a specified list after removing the 0th, 4th and 5th elements.**

Answer:-

```
my_list = [10, 20, 30, 40, 50, 60, 70, 80]
del my_list[5]
del my_list[4]
del my_list[0]
print("List after removing the 0th, 4th, and 5th elements:", my_list)
```

Output:-

```
19
List after removing the 0th, 4th, and 5th elements: [20, 30, 40, 70, 80]
```

9. **Write a Python program to generate and print a list of first and last 5 elements where the values are square of numbers between 1 and 30**

Answer:-

```
squares = []
for i in range(1, 31):
squares.append(i**2
 )
print("First 5 elements:", squares[:5])
print("Last 5 elements:", squares[-5:])
```

Output:-

```
First 5 elements: [1, 4, 9, 16, 25]
Last 5 elements: [676, 729, 784, 841, 900]
```

10. **Write a Python Program to print all Possible Combinations from the three Digits**

Answer:-

```
a = int(input("Enter first digit: "))
b = int(input("Enter second digit: "))
c = int(input("Enter third digit: "))
digits = [a, b, c]
for i in range(3):
for j in range(3):
for k in range(3):
if i != j and j != k and k != i:
print(digits[i],digits[j],digits[k])
```

Output :-

```
Enter first digit: 123
Enter second digit: 23
Enter third digit: 5
123 23 5
```

```
123 5 23
23 123 5
23 5 123
5 123 23
5 23 123`
```

Name:- Subhasis Sahoo

Rollno-01

sic-23mmci50

Branch-MCA

Group-A1

Assignment:-6
DICTIONARY

1. **Merge following two Python dictionaries into one**

Answer:-

```
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}
dict1.update(dict2)
print(dict1)
```

Output:-

```
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

2. **Get the key of a minimum value from the following dictionary**

Answer:-

```
dict = {'a': 10, 'b': 20, 'c': 5, 'd': 15}
min_key = sorted(dict, key=dict.get)[0]
print(min_key)
```

Output:-

```
c
```

3. **Write a Python script to concatenate following dictionaries to create a new One. Sample Dictionary :**

dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

Answer:-

```
dic1 = {1: 10, 2: 20}
dic2 = {3: 30, 4: 40}
dic3 = {5: 50, 6: 60}
result = dic1.copy()
result.update(dic2)
result.update(dic3) print(result)
```

Output:-

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

4. **Write a Python script to check whether a given key already exists in a dictionary**

Answer:-

```
dict = {1: 'apple', 2: 'banana', 3: 'cherry'}
key = 2
if key in dict:
print("Key
exists.")
else:
print("Key does not exist.")
```

Output:-

```
Key exists.
```

5. **Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x)**

Sample Dictionary

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13:
169, 14: 196, 15: 225}

Answer:-

```
n = 15
dict = {}
for x in range(1, n+1):
dict[x] = x * x
print(dict)
```

Output:-

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144,
13: 169, 14:
196, 15: 225}
22
```

6. **Write a Python program to sum all the items in a dictionary**

Answer:-

```
dict = {'a': 10, 'b': 20, 'c': 30, 'd': 40}
total_sum =
sum(dict.values())
print(total_sum)
```

Output:-

```
100
```

7. **Write a Python program to multiply all the items in a dictionary.**

Answer:-

```
dict = {'a': 2, 'b': 3, 'c': 4, 'd': 5}
result = 1
for value in
dict.values(): result
*= value
print(result)
```

Output:-

```
120
```

8. **Write a Python program to get the maximum and minimum value in a dictionary.**

Answer:-

```
dict = {'a': 10, 'b': 20, 'c': 5, 'd': 15}
max_value = max(dict.values())
min_value = min(dict.values())
print("Maximum value:",
max_value) print("Minimum
value:", min_value)
```

Output:-

```
Maximum value: 20
Minimum value: 5
```

9. **Write a Python program to remove duplicates from dictionary**

Answer:-

```
dict = {'a': 10, 'b': 20, 'c': 10, 'd': 30, 'e': 20}
unique_dict = {}
seen_values = set()
for key, value in
dict.items(): if value not
in seen_values:
unique_dict[key] =
value
seen_values.add(value)
print(unique_dict)
```

Output:-

```
{'a': 10, 'b': 20, 'd': 30}
```

10. **Write a Python program to combine two dictionary adding values for common keys .**

d1 = {'a': 100, 'b': 200, 'c':300}

Answer:-

```
d1 = {'a': 100, 'b': 200, 'c': 300}
d2 = {'a': 300, 'b': 200, 'd': 400}
combined_dict =
d1.copy() for key, value in
d2.items():
if key in combined_dict:
combined_dict[key] += value
else:
combined_dict[key] = value
print(combined_dict)
```

Output:-

```
{'a': 400, 'b': 400, 'c': 300, 'd': 400}
```

11. **Write a Python program to get the top three items in a shop**

Sample data: {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24}

Expected Output:

```
item4 55
item1 45.5
item3 41.3
```

Answer:-

```
shop = {'item1': 45.50, 'item2': 35, 'item3': 41.30, 'item4': 55, 'item5': 24}
sorted_items = sorted(shop.items(), key=lambda x: x[1],
reverse=True) top_three_items = sorted_items[:3]
for item, price itop_three_items:
print(item, price)
```

Output:-

```
item4 55
item1 45.5
item3 41.3
```

Name:- Subhasis Sahoo

Rollno-01

sic-23mmci50

Branch-MCA

Group-A1

Assignment – 7
(Set)

1. **Add a list of elements to a set.**

```
thisset = {"apple", "banana", "cherry"}

thisset.add("orange")

thisset.add("orange")

print(thisset)
```

Output

```
{'orange', 'apple', 'banana', 'cherry'}
```

2. **Return a new set of identical items from two sets**

ANS:-

```
thisset1 = {"apple", "banana", "cherry"}

thisset2 = {"apple", "watermilon", "cherry"}

thisSet3 = thisset1.intersection(thisset2)

print(thisSet3)
```

Output

```
{'apple', 'cherry'}
```

3. **Write a Python program to return a new set with unique items from both sets by removing duplicates.**

ANS:-

```
thisset1 = {"apple", "banana", "cherry"}

thisset2 = {"apple", "watermilon", "cherry"}

thisSet3 = thisset1.union(thisset2)

print(thisSet3)
```

Output

```
{'apple', 'banana', 'watermilon', 'cherry'}
```

4. Update the first set with items that don’t exist in the second set.

ANS:-

```
set1 = {1, 2, 3, 4, 5}

set2 = {4, 5, 6, 7}

set1.update(set2.difference(set1))

print(set1)
```

Output

```
{1, 2, 3, 4, 5, 6, 7}
```

5. Update the first set with items that don’t exist in the second set.

ANS:-

```
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}
set1 = set1 - set2
print(set1)
```

Output:-

```
{1, 2, 3}
```

6. Update the first set with items that don’t exist in the second set.

ANS:-

```
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7}
commonElements = set1.intersection(set2)
res = set()
res.update(set1.difference(commonElements))
res.update(set2.difference(commonElements))
print(res)
```

Output-

```
{1, 2, 3, 6, 7}
```

7. Check if two sets have any elements in common. If yes, display the common elements.

ANS:-

```
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7}
diff1 = set1.difference(set2)
diff2 = set2.difference(set1)
```

```
res = diff1.union(diff2)
print(res)
```

Output-

```
{1, 2, 3, 6, 7}
```

8. Update set1 by adding items from set2, except common items.

ANS:-

```
set1 = {1, 2, 3, 4, 5}

set2 = {4, 5, 6, 7}

diff1 = set1.difference(set2)

set1 = set1.difference(diff1)

print(set1)
```

Output

```
{4,5}
```

9. Remove items from set1 that are not common to both set1 and set2

ANS:-

```
set1 = {11, 2, 30, 4, 5}
print("Max: ", max(set1))
print("Min: ", min(set1))
```

Output-

```
Max:  30
Min:  2
```

10. Write a Python program to find maximum and the minimum value in a set

ANS:-

```
set1 = {1, 2, 3}
set2 = {4, 5, 6, 7}
res = set1.intersection(set2)
if  res:
   print("Have common elements")
else:
   print("No common elements")
```

Output

```
No common elements
```

Name:- Subhasis Sahoo

Rollno-01

sic-23mmci50

Branch-MCA

Group-A1

Assignment-8
Function

1. **Write a Python function to find the Max of three numbers**
   ANS:-

```
def get_max(a, b, c):
   return max(a, b, c)
a = int(input("Enter 1st No. : "))
b = int(input("Enter 2nd No. : "))
c = int(input("Enter 3rd No. : "))
print("Max number:", get_max(a, b, c))
```

Output

```
Enter 1st No. : 10
Enter 2nd No. : 30
Enter 3rd No. : 5
Max number: 30
```

2. Write a Python function to sum all the numbers in a list
   ANS:-

```
print("Enter list elements: ")
List = [int(x) for x in input().split()]
print(List)
def sum(List):
   n = len(List)
   sum = 0
   for i in List:
      sum += i
   return sum
print("Sum of elements: ", sum(List))
```

Output

```
Enter list elements:
50 60 42 87 63 15 34
[50, 60, 42, 87, 63, 15, 34]
Sum of elements:  351
```

3. **Write a Python function to check whether a number falls in a given range.**
   ANS:-

```
def given_range(n):
   if 17<n<66:
      print("input number is given Range")
   else:
      print("input number is given Out of Range")
 n = int(input("Enter number: "))
given_range(n)
```

Output:-

```
Enter number: 25

input number is given Range
```

4. **Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.**

ANS:-

```
def count_case(my_string):
   u, l = 0, 0
   for i in my_string:
      if i.isupper():
         u = u + 1
      if i.islower():
         l = l + 1
   return u, l

str1 = input("Enter String: ")
upper, lower = count_case(str1)
print("Upper:", upper)
print("Lower:", lower)
```

Output

```
Enter String: Anil SAhu
Upper: 3
Lower: 5
```

5. **Write a Python function that takes a list and returns a new list with unique elements of the first list**

ANS:-

```
print("Enter list elements: ")
List = [int(x) for x in input().split()]
print(List)
def unique_list(myList):
   list1 = list(set(myList))
   return list1
print("Unique List: ",unique_list(List))
```

Output

```
Enter list elements:
10 52 41 10 52 0 86
[10, 52, 41, 10, 52, 0, 86]
Unique List:  [0, 41, 10, 52, 86]
```

6. **Write a Python function that takes a number as a parameter and check the number is prime or not**

ANS:-

```
import math
def is_prime(n):
```

```
            if n <= 1:
                return False
            if n == 2:
                return True
            for i in range(2, int(math.sqrt(n)) + 1):
                if n % i == 0:
                    return False
return True

        n = int(input("Enter number: "))
        if is_prime(n):
            print("Prime")
        else:
            print("Not Prime")
```

```
Output
Enter number: 5
Prime
```

7. **Write a Python function that checks whether a passed string is palindrome or not**
   ANS:-

```
def is_palindrom(str1):
    return str1==str1[::-1]
s = input("Enter String: ")
if is_palindrom(s):
    print("palindrom")
else:
    print("not palindrom")
```

```
Output
Enter String: madam
Palindrom
```

8. **Create a function showEmployee() in such a way that it should accept employee name, and its salary and display both. If the salary is missing in the function call assign default value 9000 to salary.**

ANS:-

```
def emp_info(name, sal=9000):

    print("Employee Name: ",name)

    print("Employee Salary: ",sal)

name, sal = input("Enter Name & Salary :").split()

emp_info(name,sal)

emp_info(name)
```

Output

```
Enter Name & Salary :Anil 22540

Employee Name:  Anil

Employee Salary:  22540

Employee Name:  Anil

Employee Salary:  9000
```

9. **Python program to display all integers within the range 100-200 whose sum of digits is an even number using function.**
   ANS:-

```
def sum_digits(n):
   sum = 0
   while n>0:
      sum = sum + n%10
      n = n//10
   return sum
for i in range(100,200+1):
   if(sum_digits(i)%2==0):
      print(i, end=' ')
```

Output

```
101 103 105 107 109 110 112 114 116 118 121 123 125 127 129 130 132 134 136 138
141 143 145 147 149 150 152 154 156 158 161 163 165 167 169 170 172 174 176
178 181 183 185 187 189 190 192 194 196 198 200
```

10. **Python program to generate the prime numbers from 1 to N**

    ANS:-

```
import math
def is_prime(n):
   if n <= 1:
      return False
   if n == 2:
      return True
   for i in range(2, int(math.sqrt(n)) + 1):
      if n % i == 0:
         return False
   return True

s,e = input("Enter number: ").split()

for i in range(int(s), int(e)+1):
    if is_prime(i):
        print(i, end=" ")
```

Output

```
Enter number: 50 150
53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149
```

Name:- Subhasis Sahoo

Rollno-01

sic-23mmci50

Branch-MCA

Group-A1

Assignment – 9

OOP

1. **Write a Python program to create a person class. Include attributes like name, country and date of birth. Implement a method to determine the person's age.**

   Answer:-

```
   from datetime import date
   class Person:
  def __init__(self,name,country,dob):
    self.name = name
    self.country = country
    self.dob = dob
  def getAge(self, birthDate):
    today = date.today()
    age = today.year - birthDate.year
    return age
  def display(self):
    print("Name = ",self.name)
    print("Country = ",self.country)
    print("DOB = ",self.dob)


obj = Person("Asish","India",date(2002,10,10))
obj.display()
print("Age = ",obj.getAge(date(2002,10,10)), "years")
```

Output:-

```
Name =  Asish
Country =  India
DOB =  2002-10-10
Age =  22 years
```

2. Write a Python program to create a class representing a bank. Include methods for managing customer accounts and transactions

Answer:-

```
class Bank:
    def __init__(self, accNo, name, bal):
        self.account_number = accNo
        self.name = name
        self.balance = bal

    def display(self):
        print("Acc No. ",self.account_number)
        print("Name: ",self.name)
        print("Balance: ",self.balance)

    def deposite(self, amount):
        self.balance += amount

    def withdraw(self, amount):
        if amount<=self.balance:
            self.balance += amount
        else:
            print("Insufficient balance")

obj = Bank("SBIN99146500", "Anil", 1250)
obj.display()
print()
obj.deposite(5000)
obj.display()
print()
obj.withdraw(800)
obj.display()
```

```
Output:-
Acc No.  SBIN99146500
Name:  Anurag
Balance:  1250
Acc No.  SBIN99146500
Name:  Anurag
Balance:  6250
Acc No.  SBIN99146500
Name:  Anurag
Balance:  7050
```

3. **Write a Python program to create a class representing a shopping cart. Include methods for adding and removing items, and calculating the total price.**

Answer:-

```
class ShoppingCart:
   def __init__(self):
            self.items = []
   def add_item(self, name, price):
             self.items.append({'name': name, 'price': price})
             print(f"{name} added to the cart.")
   def remove_item(self, name):
      for item in self.items:
         if item['name'] == name:
            self.items.remove(item)
            print(f"{name} removed from the cart.")
            return
      print(f"{name} not found in the cart.")
   def calculate_total(self):
      total = sum(item['price'] for item in self.items)
      return total
   def show_cart(self):
      if not self.items:
            print("Your cart is empty.")
        else:
            print("Items in your cart:")
            for item in self.items:
                print(f"- {item['name']}: ${item['price']:.2f}")
cart = ShoppingCart()
cart.add_item("Laptop", 999.99)
cart.add_item("Mouse", 49.99)
cart.show_cart()
print(f"Total: ${cart.calculate_total():.2f}")
cart.remove_item("Mouse")
cart.show_cart()
print(f"Total: ${cart.calculate_total():.2f}")
```

Output:-

```
Laptop added to the cart.
Mouse added to the cart.
Items in your cart:
Laptop: $999.99
Mouse: $49.99
Total: $1049.98
Mouse removed from the cart.
Items in your cart:
Laptop: $999.99
Total: $999.99
```

4. Write a Python program to create a calculator class. Include methods for basic arithmetic class calculator:

```
    def __init__(self,a,b):
        self.a = a
        self.b = b
       def add(self):
```

```
   return self.a + self.b
    def sub(self):
        return self.a - self.b
    def mul(self):
        return self.a * self.b
    def div(self):
        return self.a // self.b
obj = calculator(10,5)
print("Sum = ",obj.add())
print("Substration = ",obj.sub())
print("Multiplication = ",obj.mul())
print("Divide = ",obj.div())
```

Output:-

```
Sum =  15
Substration =  5
Multiplication =  50
Divide =  2
```

5. **Write a Python function student_data () that will print the ID of a student (student_id). If the user passes an argument student_name or student_class the function will print the student name and class.**

Answer:-

```
class Student:
    def student_data(self, student_id, student_name=None, student_class=None):
        print("Student ID:", student_id)
        if student_name:
            print("Student Name:", student_name)
        if student_class:
            print("Student Class:", student_class)
obj = Student()
```

```
obj.student_data("23mmci24")
print()
obj.student_data("23mmci24", "Anurag")
print()
obj.student_data("23mmci76", "Payal", "MCA")
```

Output:-

```
Student ID: 23mmci24
Student Name: Anurag
Student ID: 23mmci76
Student Name: Payal
Student Class: MCA
```

Name:- Subhasis Sahoo

Rollno-01

sic-23mmci50

Branch-MCA

Group-A1

Assignment-10

PANDAS & DATAFRAME

1. **Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.**

CODE :

```
import pandas as pd
data = {
'Name': ['John', 'Alice', 'Bob'],
'Age': [25, 30, 22],
'City': ['New York', 'Los Angeles', 'Chicago']
}
index_labels = ['A', 'B', 'C']
df = pd.DataFrame(data, index=index_labels)
print(df)
```

OUTPUT :

```
Name  Age       City
A  John   25     New York
B Alice   30  Los Angeles
C   Bob   22      Chicago
```

2. **Write a Pandas program to get the first 3 rows of a given DataFrame.**

CODE :

```
import pandas as pd
data = {
'Name': ['John', 'Alice', 'Bob', 'Eve', 'Charlie'],
'Age': [25, 30, 22, 28, 35],
'City': ['New York', 'Los Angeles', 'Chicago', 'Houston', 'Phoenix']
}
df = pd.DataFrame(data)
first_three_rows = df.head(3)
print(first_three_rows)
```

OUTPUT :

```
47 | P a g e
Name    Age     City
0   John   25     New York
1  Alice   30     Los Angeles
2    Bob   22      Chicago
```

3. Write a Pandas program to select the ‘name’ and ‘score’ columns from the following DataFrame.

Note: sample python Dictionary data and list labels:

```
exam_data = { 'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael',
'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes'] }
```

CODE :

```
import pandas as pd
import numpy as np
exam_data = {
'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael',
'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']
}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

```
df = pd.DataFrame(exam_data, index=labels)
selected_columns = df[['name', 'score']]
print("Select specific columns:")
print(selected_columns)
```

OUTPUT :

```
Select specific columns:
name  score
a  Anastasia   12.5
b      Dima    9.0
c  Katherine   16.5
d     James    NaN
e     Emily    9.0
f   Michael   20.0
g   Matthew   14.5
h     Laura    NaN
i    Kevin    8.0
j    Jonas   19.0
```

4. **Write a Pandas program to select the rows where the number of attempts in the examination is greater than 2**

CODE :

```
import pandas as pd
import numpy as np
exam_data = {
'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura',
'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']
}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

```
df = pd.DataFrame(exam_data, index=labels)

filtered_rows = df[df['attempts'] > 2]

print(filtered_rows)
```

OUTPUT :

```
name  score  attempts qualify

b    Dima    9.0         3      no

d   James    NaN       3      no

f  Michael   20.0       3     yes
```

5. **Write a Pandas program to select the rows where the score is missing, i.e. is NaN**

CODE :

```
import pandas as pd

import numpy as np

exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data, index=labels)

filtered_rows = df[df['score'].isna()]

print(filtered_rows)
```

OUTPUT :

```
name  score      attempts    qualify

d  James    NaN        3         no

h  Laura    NaN        1         no
```

6. **Write a Pandas program to select the rows the score is between 15 and 20 (inclusive).**

CODE :

```
import pandas as pd

import numpy as np

exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts':
```

```
[1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data, index=labels)
filtered_rows = df[(df['score'] >= 15) & (df['score'] <= 20)]
print(filtered_rows)
```

OUTPUT :

```
name  score  attempts qualify
c  Katherine   16.5         2      yes
 f   Michael   20.0          3       yes
 j    Jonas   19.0              1     yes
```

7. **Write a Pandas program to select the rows where number of attempts in the examination is less than 2 and score greater than 15**

CODE :

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts':
[1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data, index=labels)
filtered_rows = df[(df['attempts'] < 2) & (df['score'] > 15)]
print(filtered_rows)
```

OUTPUT :

```
    name      score  attempts qualify
f  Michael   20.0         1        yes
```

8. **Write a Pandas program to calculate the sum of the examination attempts by the students.**

CODE :

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael',   'Matthew',
'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts':
[1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data, index=labels)
total_attempts = df['attempts'].sum()
print("Total number of examination attempts:", total_attempts)
```

OUTPUT :

```
Total number of examination attempts: 18
```

9. **Write a Pandas program to append a new row 'K'to data frame with given values for each column. Now delete the new row and return the original DataFrame.**

CODE :

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no',
'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data, index=labels)
new_row = {'name': 'Chris', 'score': 18, 'attempts': 2, 'qualify': 'yes'}
df.loc['k'] = new_row
df = df.drop('k')
print(df)
```

OUTPUT :

```
name  score  attempts qualify


a  Anastasia   12.5      1   yes
 b  Dima        9.0       3    no
 c  Katherine   16.5    2    yes
 d     James     NaN     3     no
```

```
e     Emily    9.0        2      no
f   Michael   20.0       3      yes
g   Matthew   14.5      1      yes
h     Laura    NaN      1      no
i     Kevin     8.0        2       no
j     Jonas    19.0        1       yes
```

10. Write a Pandas program to sort the DataFrame first by &#39;name&#39; in descending order,then by 'score' in ascending order.

CODE :

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts':
[1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data, index=labels)
sorted_df = df.sort_values(by=['name', 'score'], ascending=[False, True])
print(sorted_df)
```

OUTPUT :

```
name  score  attempts qualify
c  Katherine   16.5    2     yes
j     Jonas       19.0    1     yes
f   Michael   20.0      3     yes
b      Dima    9.0        3     no
a  Anastasia   12.5     1     yes
e     Emily    9.0        2      no
g   Matthew   14.5     1      yes
i     Kevin    8.0        2       no
h     Laura    NaN     1       no
d     James    NaN     3       no
```

11 .Write a Pandas program to replace the &#39;qualify&#39; column contains the values ‘yes’ and ‘no’ with True and False

CODE :

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data, index=labels)
df['qualify'] = df['qualify'].replace({'yes': True, 'no': False})
print(df)
```

OUTPUT :

```
name          score     attempts  qualify
a  Anastasia   12.5       1           True
b      Dima      9.0       3           False
c  Katherine   16.5       2           True
d     James    NaN        3           False
e     Emily    9.0        2           False
f   Michael   20.0        3           True
g   Matthew   14.5        1          True
h     Laura    NaN        1           False
i    Kevin    8.0          2          False
j    Jonas   19.0          1          True
```

12 . Write a Pandas program to delete the &#39;attempts&#39; column from the DataFrame.

CODE :

```
import pandas as pd
import numpy as np
```

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts':
[1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data, index=labels)
df = df.drop('attempts', axis=1)
print(df)
```

OUTPUT :

```
name  score  qualify
a  Anastasia   12.5     yes
b      Dima    9.0      no
c  Katherine   16.5     yes
d     James    NaN      no
e     Emily    9.0      no
f   Michael   20.0     yes
g   Matthew   14.5     yes
h     Laura    NaN      no
i     Kevin    8.0      no
j     Jonas   19.0     yes
```

Name:- Subhasis Sahoo

Rollno-01

sic-23mmci50

Branch-MCA

Group-A1

ASSIGNMENT-11

SQLITE

1.Write a Python program to create a SQLite database and connect with the database and print the version of the SQLite database.

Answer:

```
import sqlite3
con=sqlite3.connect("sqlite.db")
print("the version of the SQLite database is ",sqlite3.version) con.close()
```

output:

```
the version of the SQLite database is 2.6.0
```

2.Write a Python program to create a SQLite database connection to a database that resides in the memory.

Ans:

```
import sqlite3
conn=sqlite3.connect('sqlite. db') print(conn)
conn.close()
```

Output:

```
<sqlite3.Connection object at 0x0000022D4EB75210>
```

3.Write a Python program to connect a database and create a SQLite table within the database.

Ans:

```
import sqlite3
con=sqlite3.connect("sqlite.db")
qry="CREATE TABLE demo(id AUTO_INCRIMENT PRIMARY KEY,name
VARCHAR(20))"
res=con.execute(qry) print("Table created.") con.close()
```

Output:

```
Table created.
```

4.Write a Python program to list the tables of given SQLite database file. Ans:

```
import sqlite3
con=sqlite3.connect("sqlite.db")
qry="SELECT name FROM sqlite_master WHERE type='table';" res=con.execute(qry)
print(res.fetchall()) con.close()
```

Output:

```
[('demo',)]
```

5.Write a Python program to create a table and insert some records in that table. Finally selects all rows from the table and display the records.

Ans:

```
import sqlite3
con=sqlite3.connect("sqlite.db")
```

```
qry="INSERT INTO demo(name) VALUES(?)" val=[
("himansu",),("soumendra",),("subhasis",),("xyz",)
]
con.executemany(qry,val) con.commit()
res=con.execute("SELECT * FROM demo").fetchall() for i in res:
print(i[1]) con.close()
```

Output:

```
himansu
soumendra subhasis xyz
```

6.Write a Python program to insert a list of records into a given SQLite table.

Ans:

```
mport sqlite3
con=sqlite3.connect("sqlite.db")
qry="INSERT INTO demo(name) VALUES(?)" val=[
("himansu",),("soumendra",),("subhasis",),("xyz",)

]
con.executemany(qry,val) con.commit()
con.close()
```

Output:

```
Process finished with exit code 0
```

7.Write a Python program to insert values to a table from user input.

Ans:

```
import sqlite3
con=sqlite3.connect("sqlite.db")
qry="INSERT INTO demo(id,name) VALUES(?,?)" val=[]
n=int(input("how many students are their: ")) print("Enter the id and names of the students: ")
for i in range(0,n):
val.append((input(),input())) con.executemany(qry,val)
con.commit() con.close()
```

Output:

```
how many students are their: 1
Enter the id and names of the students:
2
Manjayee
```

8.Write a Python program to count the number of rows of a given SQLite table. Ans:

```
import sqlite3
con=sqlite3.connect("sqlite.db") qry="SELECT COUNT(*) FROM demo"
res=con.execute(qry).fetchone()[0]
print("The number of rows in the table are: ",res) con.close()
```

Output:

```
The number of rows in the table are: 9
```

9.Write a Python program to update a specific column value of a given table and select all rows before and after updating the said table.

Ans:

```
import sqlite3
con=sqlite3.connect("sqlite.db") def getRow():
res=con.execute("SELECT * FROM demo").fetchall() print("Id Name")
for i in res: print(i[0],i[1])

print("Table Data before Update") getRow()
id=input("Enter the student id you want to update: ")
name=input("Enter the student name you want to update: ") qry="UPDATE demo SET
name='"+name+"' where id="+id print(qry)
con.execute(qry)
print("Table Data after Update") getRow()
```

Output:

```
Table Data before Update Id Name
None himansu None soumendra None subhasis None xyz
None himansu None soumendra

None subhasis None xyz
2 manjayee
Enter the student id you want to update: 2
Enter the student name you want to update: smruti UPDATE demo SET name='smruti' where
id=2
Table Data after Update Id Name
None himansu None soumendra None subhasis None xyz
None himansu None soumendra None subhasis None xyz
2 smruti
```

10. Write a Python program to update all the values of a specific column of a given SQLite table.

Ans:

```
import sqlite3
con=sqlite3.connect("sqlite.db") def getRow():
res=con.execute("SELECT * FROM demo").fetchall() print("Id Name")
for i in res: print(i[0],i[1])

print("Table Data before Update") getRow()

qry="UPDATE demo SET name='xyz'" print(qry)
con.execute(qry)
print("Table Data after Update") getRow()
```

Output:

```
Table Data before Update Id Name
None himansu None soumendra None subhasis None xyz
None himansu None soumendra None subhasis None xyz
2 manjayee
```

```
UPDATE demo SET name='xyz' Table Data after Update
Id Name None xyz None xyz None xyz None xyz None xyz None xyz None xyz None xyz 2
xyz
```

Name:- Subhasis Sahoo

Rollno-01

sic-23mmci50

Branch-MCA

Group-A1

ASSIGNMENT – 12

```
data = {
        birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers',
        'Cranes', 'spoonbills', 'spoonbills'],

        'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],

        'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2], 'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no',
        'yes', 'no', 'no']
}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

1. Create a DataFrame birds from this dictionary data which has the index labels.

Answer-

```
#Create a DataFrame birds from this dictionary data which has the index labels.
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2], 'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no',
'no']
}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
birds = pd.DataFrame(data,index=labels)
print(birds)
```

Output:-

```
        birds  age  visits priority
a      Cranes  3.5       2      yes
b      Cranes  4.0       4      yes
c     plovers  1.5       3       no
d  spoonbills  NaN       4      yes
e  spoonbills  6.0       3       no
f      Cranes  3.0       4       no
g     plovers  5.5       2       no
h      Cranes  NaN       2      yes
```

```
i  spoonbills  8.0       3       no
j  spoonbills  4.0       2       no
```

2. Display a summary of the basic information about birds DataFrame and its data.

Answer-

```
#Display a summary of the basic information about birds DataFrame and its data.
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2], 'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no',
'no']
}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
birds = pd.DataFrame(data,index=labels)
print("Basic Information:")
print(birds.info())
print("\nDescriptive Statistics:")
print(birds.describe())
print("\nFirst Few Rows:")
print(birds.head())
```

Output:-

```
Basic Information:
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   birds     10 non-null     object
 1   age       8 non-null      float64
 2   visits    10 non-null     int64
 3   priority  10 non-null     object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
None

Descriptive Statistics:
            age     visits
count  8.000000  10.000000
mean   4.437500   3.000000
std    2.010360   0.816497
min    1.500000   2.000000
25%    3.000000   2.250000
50%    4.000000   3.000000
```

```
75%    5.875000   3.750000
max    8.000000   4.000000

First Few Rows:
     birds  age  visits priority
a   Cranes  3.5       2      yes
b   Cranes  4.0       4      yes
c  plovers  1.5       3       no
d spoonbills  NaN       4      yes
e spoonbills  6.0       3       no
```

3. Print the first 2 rows of the birds dataframe

Answer-

```
#Print the first 2 rows of the birds dataframe
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2], 'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no',
'no']
}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
birds = pd.DataFrame(data,index=labels)
print("\nFirst Two Rows:")
print(birds.head(2))
```

Output:-

```
First Two Rows:
    birds  age  visits priority
a  Cranes  3.5       2      yes
b  Cranes  4.0       4      yes
```

4. Print all the rows with only 'birds' and 'age' columns from the dataframe

Answer-

```
#Print all the rows with only 'birds' and 'age' columns from the dataframe
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2], 'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no',
'no']
}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
birds = pd.DataFrame(data,index=labels)
print(birds[['birds','age']])
```

Output:-

```
       birds  age
a     Cranes  3.5
b     Cranes  4.0
c    plovers  1.5
d  spoonbills  NaN
e  spoonbills  6.0
f     Cranes  3.0
g    plovers  5.5
h     Cranes  NaN
i  spoonbills  8.0
j  spoonbills  4.0
```

5. select [2, 3, 7] rows and in columns ['birds', 'age', 'visits']

Answer-

```
#select [2, 3, 7] rows and in columns ['birds', 'age', 'visits']
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2],
'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']
}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

birds = pd.DataFrame(data,index=labels)
# select = birds.iloc[[2,3,7],['birds','age','visits']]
select = birds.iloc[[2, 3, 7], [0, 1, 2]]
print(select)
```

Output:-

```
       birds  age  visits
c    plovers  1.5       3
d  spoonbills  NaN       4
```

```
h      Cranes  NaN       2
```

6. select the rows where the number of visits is less than 4

Answer-

```
#select the rows where the number of visits is less than 4
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2],
'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']
}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

birds = pd.DataFrame(data,index=labels)

select = birds[birds['visits']<4]
print(select)
```

Output:-

```
         birds  age  visits priority
a       Cranes  3.5       2      yes
c      plovers  1.5       3       no
g      plovers  5.5       2       no
h       Cranes  NaN       2      yes
i   spoonbills  8.0       3       no
j   spoonbills  4.0       2       no
```

7. select the rows with columns ['birds', 'visits'] where the age is missing i.e NaN

Answer-

```
#select the rows with columns ['birds', 'visits'] where the age is missing i.e NaN
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2],
'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']
}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

```
birds = pd.DataFrame(data,index=labels)
select = birds[birds['age'].isna()][['birds', 'visits']]

print(select)
```

Output:-

```
      birds  visits
d  spoonbills      4
h      Cranes      2
```

8. Select the rows where the birds is a Cranes and the age is less than 4

Answer-

```
# Select the rows where the birds is a Cranes and the age is less than 4
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2],
'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']
}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

birds = pd.DataFrame(data,index=labels)
select = birds[(birds['birds'] == 'Cranes')&(birds['age']<4)]
print(select)
```

Output:-

```
    birds  age  visits priority
a  Cranes  3.5       2      yes
f  Cranes  3.0       4       no
```

9. Select the rows the age is between 2 and 4(inclusive)

Answer-

```
# Select the rows the age is between 2 and 4(inclusive)
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2],
'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']
}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

birds = pd.DataFrame(data,index=labels)
select = birds[(birds['age']>=2)&(birds['age']<=4)]

print(select)
```

Output:-

```
   birds  age  visits priority
a  Cranes  3.5      2      yes
f  Cranes  3.0      4       no
```

10.Find the total number of visits of the bird Cranes

Answer-

```
# Find the total number of visits of the bird Cranes
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2],
'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']
}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

birds = pd.DataFrame(data,index=labels)
select = birds[birds['birds']=='Cranes']['visits']

print(select.sum())
```

Output:-

```
12
```

11.Calculate the mean age for each different birds in dataframe.

Answer-

```
# Calculate the mean age for each different birds in dataframe.
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2],
'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']
}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

birds = pd.DataFrame(data,index=labels)
select = birds.groupby('birds')['age']


print((select.mean()))
```

Output:-

```
birds
Cranes       3.500000
plovers      3.500000
spoonbills   6.000000
Name: age, dtype: float64
```

12.Append a new row 'k' to dataframe with your choice of values for each column. Then delete that row to return the original DataFrame.

Answer-

```
#Append a new row 'k' to dataframe with your choice of values for each column. Then delete
#that row to return the original DataFrame
import pandas as pd
import numpy as np
data = {
'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills',
'spoonbills'],
'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2],
'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']
}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
k = [1,2,3,4,5,6,7,8,9,0]
```

```
data['k'] = k
birds = pd.DataFrame(data,index=labels)
# birds.remove(birds['k'])
print(birds)
```

Output:-

```
DataFrame after adding row 'k':
        birds  age  visits priority
a      Cranes  3.5       2      yes
b      Cranes  4.0       4      yes
c     plovers  1.5       3       no
d  spoonbills  NaN       4      yes
e  spoonbills  6.0       3       no
f      Cranes  3.0       4       no
g     plovers  5.5       2       no
h      Cranes  NaN       2      yes
i  spoonbills  8.0       3       no
j  spoonbills  4.0       2       no
k     Eagles  5.5       3      yes

DataFrame after deleting row 'k':
        birds  age  visits priority
a      Cranes  3.5       2      yes
b      Cranes  4.0       4      yes
c     plovers  1.5       3       no
d  spoonbills  NaN       4      yes
e  spoonbills  6.0       3       no
f      Cranes  3.0       4       no
g     plovers  5.5       2       no
h      Cranes  NaN       2      yes
i  spoonbills  8.0       3       no
j  spoonbills  4.0       2       no
```

Name:- Subhasis Sahoo

Rollno-01

sic-23mmci50

Branch-MCA

Group-A1