

Q1. Write a program to understand the different basic data type.

Program:

```
num1 = 5
print(num1, 'is of type', type(num1))
num2 = 2.0
print(num2, 'is of type', type(num2))
num3 = 1+2j
print(num3, 'is of type', type(num3))
str1='Hello'
print(str1, 'is of type', type(str1))
```

Output:

```
5 is of type <class 'int'>
2.0 is of type <class 'float'>
(1+2j) is of type <class 'complex'>
Hello is of type <class 'str'>
```

Q2. Write a program to read two integers and find the sum, diff, multiply and div.

Program:

```
a=int(input("enter first number: "))
b=int(input("enter second number: "))
c=a+b
d=a-b
e=a/b
f=a%b
g=a*b
print("sum: ",c)
print("difference: ",d)
print("multiplication: ",g)
print("division: ",e)
print("remainder: ",f)
```

Output:

```
enter first number: 4
enter second number: 7
sum: 11
difference: -3
multiplication: 28
division: 0.5714285714285714
remainder: 4
```

Q3. Write a program to find the simple interest for a given value P, T and R. The program must take the input from the user.

Program:

```
p=int(input("Enter Principal Amount : "))
r=int(input("Enter Rate of Interest: "))
t=int(input("Enter Time: "))
s=(p*r*t)/100
print("simple interest is: ",s)
```

Output:

```
Enter Principal Amount : 1000
Enter Rate of Interest: 5
Enter Time: 4
simple interest is: 200.0
```

Q4. Write a program to show the use of operators in Python

Program:

```
a=int(input("Enter first number: "))
b=int(input("Enter second number: "))
print("Is a>b: ",a>b)
print("Is a<b: ",a<b)
print("Is a>=b: ",a>=b)
print("Is a<=b: ",a<=b)
print("a!=b: ",a!=b)
print("the bitwise of a and b is: ",a&b)
print("the bitwise of a or b is: ",a|b)
print("the bitwise of a xor b is: ",a^b)
# logical AND
print(True and True)
print(True and False)
# logical OR
print(True or False)
print(not True)
```

Output:

```
Enter first number: 4
Enter second number: 5
Is a>b: False
Is a<b: True
Is a>=b: False
Is a<=b: True
a!=b: True
the bitwise of a and b is: 4
the bitwise of a or b is: 5
the bitwise of a xor b is: 1
True
False
```

True

False

Q5. Write a program to find greatest of three numbers using:

A. If-elif-else ladder

B. nested if

Program A:

```
a=int(input("Enter first number: "))
b=int(input("Enter second number: "))
c=int(input("Enter third number: "))
if (a>b):
    if (a>c):
        print(a,"is largest")
    else:
        print(c,"is largest")
elif (b>a):
    if (b>c):
        print(b,"is largest")
    else:
        print(c,"is largest")
else:
    print(c,"is the largest")
```

Output A:

```
Enter first number: 3
Enter second number: 2
Enter third number: 4
4 is largest
```

Program B:

```
a=int(input("Enter first number: "))
b=int(input("Enter second number: "))
c=int(input("Enter third number: "))
if (a>b and a>c):
    print('first number',a,' is greatest')
elif (b>a and b>c):
    print('second number',b,' is greatest')
else:
    print('third number',c,' is greatest')
```

Output B:

```
Enter first number: 3
Enter second number: 4
Enter third number: 2
second number 4 is greatest
```

Q6. Write a python program to generate first n Fibonacci number and factorial of n using:

A. functions

B. recursion

Program A:

```
def recur_fibo(n):
    if n <= 1:
        return "Enter valid number!!"
    else:
        l=[0,1]
        for i in range(n-2):
            l.append(l[-1]+l[-2])
        for i in l:
            print(i,end=" ")
n=int(input("How many terms? "))
recur_fibo(n)
```

Output A:

```
How many terms? 5
0 1 1 2 3
```

Program B:

```
def recur_fibo(n):
    if n <= 0:
        return "Enter valid number!!"
    elif n==1:
        print(0)
    else:
        l=[0,1]
        for i in range(n-2):
            l.append(l[-1]+l[-2])
        for i in l:
            print(i,end=" ")
n=int(input("How many terms? "))
recur_fibo(n)
```

Output:

```
How many terms? 5
Fibonacci sequence:
0 1 1 2 3
```

Date: 1 Sep 2023

Q7. Create a class Rectangle and define two attributes length and width. Define a method to calculate the area of the rectangle.

Program:

```
class rectangle:
    def __init__(self,l,b):
        self.length=l
        self.width=b
    def area(self):
        print(f'the area of the given rectangle is
{self.length}*{self.width}=',self.length*self.width)
r=rectangle(3,4)
r.area()
```

Output:

the area of the given rectangle is 3*4= 12

Q8. Write a program to show the implementation of:

a. Default constructor

b. Parameterised constructor

c. Destructor

Program A:

```
class Employee:
    def __init__(self):
        self.name='Python'
    def display(self):
        print(self.name)
emp1=Employee()
emp1.display()
```

Output B:

Python

Program B:

```
class Add:
```

```

def __init__(self, f, s):
    self.first=f
    self.second=s
def display(self):
    self.ans=self.first+self.second
    print(self.ans)
a=Add(3,4)
a.display()

```

Output B:

7

Program C:

```

class employee:
    def __init__(self):
        print("constructor is created ")
    def __del__(self):
        print("the destructor is called and now object will be deleted")
emp1=employee()
del emp1

```

Output C:

```

constructor is created
the destructor is called and now object will be deleted

```

Q9. Create a class called NUMBERS, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers).

- a. Write an instance method called add which returns the sum of the attributes x and y.
- b. Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER.
- c. Write a static method called subtract, which takes two number objects, b and c, and returns b - c.
- d. Write a method called value which returns a tuple containing the values of x and y

Program:

```

class numbers:
    multiplier=10
    def __init__(self, x, y):
        self.x=x
        self.y=y
    def add(self):
        return self.x+self.y
    @classmethod
    def multiply(cls, a):
        return cls.multiplier*a
    @staticmethod

```

```
def subtract(b,c):
    return b-c
num=numbers(5,10)
print(num.add())
print(num.multiply(5))
num.subtract(10,5)
```

Output:

```
15
50
5
```

Date: 11 Sep 2023

Q10. Create a module named 'Arithmetic' having basic arithmetic functions. Import this module and use the functions to implement the arithmetic operations.

Program:

```
from google.colab import files
files.upload()
import calc_py
print(calc_py.add(3,4))
print(calc_py.sub(3,4))
print(calc_py.mul(3,4))
print(calc_py.mod(3,4))
calc_py
```

```
def add(a,b):
    return a+b
def sub(a,b):
    return a-b
def mul(a,b):
    return a*b
def mod(a,b):
    return a%b
```

Output:

- **calc_py.py**(text/x-python) - 124 bytes, last modified: 9/22/2023 - 100% done

Saving calc_py.py to calc_py.py

```
7
-1
12
3
```

Q11. Write a program to print the following pattern:

```
A
A B
A B C
A B C D
A B C D E
```

Program:

```
for i in range(5):
```

```

for j in range(i+1):
    print(chr(65+j),end=" ")
print()

```

Output:

```

A
A B
A B C
A B C D
A B C D E

```

Q12. Write a program using split() method to split a multiline string.

Program:

```

a='''Hello I am Himanshu.
I am pursuing Btech in AI
because AI is an emerging field'''
b=a.split()
print(b)

```

Output:

```
['Hello', 'I', 'am', 'Himanshu.', 'I', 'am', 'pursuing', 'Btech', 'in', 'AI', 'because', 'AI', 'is', 'an', 'emerging', 'field']
```

Q13. Write a program that accepts a string, remove all vowels and print the remaining string.

Program:

```

a=input("Enter the string: ")
new=''
for i in a:
    if i not in 'aeiouAEIOU':
        new=new+i
a=new
print("Rquired string is: ",a)

```

Output:

```

Enter the string: Hello World
Rquired string is:  Hll Wrld

```

Q14. Write a program to find whether a given character is present in a string or not. If it is present, print its index.

Program:

```

a=input("Enter the string: ")
new=input("Enter the character to search: ")

```



```

index=None
for i in range(len(a)):
    if a[i]==new:
        index=i
        break
if index==None:
    print("Character not found")
else:
    print("Character found at index ",index)

```

Output:

```

Enter the string: Hello
Enter the character to search: l
Character found at index 2

```

Q15. Write a program to reverse a string.

Program:

```

a=input("Enter the string: ")
a=a[::-1]
print('Reversed string: ',a)

```

Output:

```

Enter the string: Hello
Reversed string: olleH

```

Q16. Write a program to count the number of occurrences of a specific character without using count() function.

Program:

```

a=input("Enter the string: ")
new=input("Enter the character to count: ")
count=0
for i in range(len(a)):
    if a[i]==new:
        count+=1
print("Character occur ",count," times")

```

Output:

```

Enter the string: Hello
Enter the character to count: l
Character occur 2 times

```

Date: 15 Sep 2023

Q17. Create an output list which contain only even numbers present in input list.

Program:

```
l1=[1,2,3,4,5,6]
l2=[]
for i in l1:
    if i%2==0:
        l2.append(i)
print("Required list: ",l2)
```

Output:

Required list: [2, 4, 6]

Q18. Create an output list which contains squares of all numbers from 1-9.

Program:

```
l1=[i**2 for i in range(1,10)]
print(l1)
```

Output:

[1, 4, 9, 16, 25, 36, 49, 64, 81]

Q19. Create an output dict which contains only odd numbers that are present in the input list as keys & their cubes as values.

Program:

```
d1={i:pow(i,3) for i in range(10) if i%2!=0}
```

```
print(d1)
```

Ouput:

```
{1: 1, 3: 27, 5: 125, 7: 343, 9: 729}
```

Q20. Given two lists containing the names of states and their corresponding capitals. Construct a dictionary which maps the states with their respective capitals.

Program:

```
states=['Jharkhand','West Bengal','Uttarakhand','Meghalaya']
capitals=['Ranchi','Kolkata','Dehradun','Shillong']
d1=dict(zip(states, capitals))
print(d1)
```

Ouput:

```
{'Jharkhand': 'Ranchi', 'West Bengal': 'Kolkata', 'Uttarakhand': 'Dehradun', 'Meghalaya': 'Shillong'}
```

Q21. Create an output set which contains even numbers present in input list.

Program:

```
l1=[i for i in range(10)]
s1={i for i in l1 if i%2==0}
print(l1)
print(s1)
```

Output:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
{0, 2, 4, 6, 8}
```

Q22. Generate a set of prime numbers and another set of even numbers, demonstrate the result of union, intersection, difference and symmetric difference operation.

Program:

```
def isPrime(n):
    for i in range(2,n):
        if n%i==0:
            return False
    return True
s1={i for i in range(20) if isPrime(i)}
s2={i for i in range(20) if i%2==0}
print(s1)
print(s2)
print("Union of s1 and s2: ",s1.union(s2))
print("Intersection of s1 and s2: ",s1.intersection(s2))
print("difference of s1 and s2: ",s1.difference(s2))
print("symmetric difference of s1 and s2: ",s1.symmetric_difference(s2))
```

Output:

```
{0, 1, 2, 3, 5, 7, 11, 13, 17, 19}
{0, 2, 4, 6, 8, 10, 12, 14, 16, 18}
Union of s1 and s2: {0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16,
17, 18, 19}
Intersection of s1 and s2: {0, 2}
difference of s1 and s2: {1, 3, 5, 7, 11, 13, 17, 19}
symmetric difference of s1 and s2: {1, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13,
14, 16, 17, 18, 19}
```

Q23. Write a python program to:**(a) To create two new files f1 and f2****(b) To read and display the contents, count the number of lines and find the word whose count is more in f1 and f2 respectively.****(c) To create and display the file f3 which is a combination of f1 and f2.****Program A:**

```
f1=open("File1",'w')
f2=open("File2",'w')
f1.write("Hello \n I am Himanshu \n I am pursuing Btech(A.I)")
f2.write("This is a text file\n I am writing the some content in it \n
Hope you like it")
f1.close()
f2.close()
```

Program B:

```
f1=open("File1",'r')
f2=open("File2",'r')
l1=f1.readlines()
l2=f2.readlines()
print("Content of l1: ")
for i in l1: print(i.rstrip('\n'))
print("\nContent of l2: ")
for i in l2: print(i.rstrip('\n'))
f1.close()
f2.close()
print("\nNumber of lines in file 1 is ",len(l1))
print("Number of lines in file 2 is ",len(l2))
```

Output B:

Content of l1:

Hello

I am Himanshu

I am pursuing Btech(A.I)

Content of l2:

```
Hello
I am Himanshu
I am pursuing Btech(A.I)
```

```
Number of lines in file 1 is 3
Number of lines in file 2 is 3
```

Program C:

```
f1=open("File1",'r')
f2=open("File2",'r')
f3=open("File3",'w')
f3.write(f1.read())
f3.write(f2.read())
f3.close()
f2.close()
f1.close()
f3=open("File3",'r')
print(f3.read())
f3.close()
```

Output C:

```
Hello
I am Himanshu
I am pursuing Btech(A.I)This is a text file
I am writing the some content in it
Hope you like it
```

Q24. Write an Object-Oriented Python program to create two Time objects: current Time, which contains the current time; and bread Time, which contains the amount of time it takes for a bread maker to make bread. Then we'll use add Time to figure out when the bread will be done. Write the print Time function to display the time when the bread will be done by the bread maker.

Program:

Output:

Q25. Design a Python program as described below:

- Create a class called Palindrome.**
- In your Palindrome class, create a method called reverse() which takes a string argument. Your method should return the reverse of the argument as a string.**
- Create a second method in Palindrome called isPalindrome() which takes a string argument. This method should return True if the argument is a palindrome and False otherwise.**
- Write some code to test your new Palindrome class and print out results of your testing to the user. Give some consideration to what sort of strings you might want to use for your testing**

Program:

```
class Palindrome:
    def reverse(self,s):
        return s[::-1]
    def isPalindrome(self,s):
        if s==s[::-1]:
            return True
        return False
c1=Palindrome()
```

```

rev=c1.reverse("Hello")
check=c1.isPalindrome("nitin")
print(rev)
print(check)

```

Output:

```

olleH
True

```

Q26. Create a class named as Student to store the name and marks in three subjects. Use List to store the marks.

Program:

```

class Student:
    def __init__(self,n,m1,m2,m3):
        self.name=n
        self.l=[m1,m2,m3]
    def display(self):
        print("Name: ",self.name)
        print("Marks: ",self.l)
c1=Student("Himanshu",12,13,14)
c1.display()

```

Output:

```

Name: Himanshu
Marks: [12, 13, 14]

```

Q27. Create a class Employee that keeps a track of the number of employees in an organization and also stores their name, designation and salary details.

- a. Write a method called getdata to take input (name, designation, salary) from user. B
- b. . Write a method called display to print all the information of an employee.

Program:

```

class Employee:
    def getdata(self,n,d,s):
        self.name=n
        self.designation=d
        self.salary=s
    def display(self):
        print("Name: ",self.name)
        print("Designation: ",self.designation)
        print("Salary: ",self.salary)
e1=Employee()
e1.getdata("Himanshu","software Developer",12301010)
e1.display()

```

Output:

```

Name: Himanshu
Designation: software Developer
Salary: 12301010

```

Q28. Create a Python class named Circle constructed by a radius. Use a class variable to define the value of constant PI. Write two methods to be named as area and circum to compute the area and the perimeter of a circle respectively by using class variable PI.

Program:

```
class circle:
    pi=3.14
    def area(self,r):
        return circle.pi*r*r
    def circum(self,r):
        return 2*circle.pi*r
c1=circle()
print("Area: ",c1.area(7))
print("Circumference: ",c1.circum(7))
```

Output:

```
Area: 153.86
Circumference: 43.96
```

Q29. Write a program that has a class called Fraction with attributes numerator and denominator. a. Write a method called getdata to enter the values of the attributes. b. Write a method show to print the fraction in simplified form.

Program:

```
class Fraction:
    def getdata(self,n,d):
        self.numerator=n
        self.denominator=d
    def show(self):
        i=2
        while(i<=min(self.numerator,self.denominator)):
            if(self.numerator%i==0 and self.denominator%i==0):
                self.numerator/=i
                self.denominator/=i
                continue
            i+=1
        print(str(self.numerator),"/",str(self.denominator))
f1=Fraction()
f1.getdata(4,8)
f1.show()
```

Output:

Date: 22 Sep 2023

Q30. Write a program that prompts the user to enter a message. Now count and print the number of occurrences of each character.

Program:

```
a=input("Enter the message: ")
d1={}
for i in a:
    if i in d1:
        d1[i]+=1
    else:
        d1[i]=1
print("Character occurrences are: ")
for character,count in d1.items():
    print(character,": ",count)
```

Output:

```
Enter the message: Hello python
Character occurrences are:
H : 1
e : 1
l : 2
o : 2
```



```
: 1
p : 1
y : 1
t : 1
h : 1
n : 1
```

Q31. Write a program to store sparse matrix as a dictionary.

Program:

```
arr = [[0, 0, 0, 1, 0],
        [5, 0, 0, 0, 4],
        [0, 0, 0, 2, 0]]
dic = {}
for i in range(len(arr)):
    for j in range(len(arr[i])):
        if arr[i][j] != 0:
            dic[i, j] = arr[i][j]
print("Position of non-zero elements in the matrix:")
print(dic)
```

Output:

```
Position of non-zero elements in the matrix:
{(0, 3): 1, (1, 0): 5, (1, 4): 4, (2, 3): 2}
```

Q32. Write a program that inverts a dictionary. That is, it makes key of one dictionary value of another and vice versa.

Program:

```
d={'A':1,'B':2,'C':3}
print("Original dictionary: ",d)
d1={}
for i,j in d.items():
    d1[j]=i
print("Inverted dictionary: ",d1)
```

Output:

```
Original dictionary: {'A': 1, 'B': 2, 'C': 3}
Inverted dictionary: {1: 'A', 2: 'B', 3: 'C'}
```

Q33. Write a program that has dictionary of names of students and a list of their marks in 4 subjects. Create another dictionary from this dictionary that has name of the students and their total marks. Find out the topper and his/ her score.

Program:

```
d1={'Student1':[12,13,14,15],"Rishab":[14,15,11,13],"Sintu":[15,12,11,14]}
d={}
```

```

for i,j in d1.items():
    sum=0
    for k in j:
        sum+=k
    d[i]=sum
topper=''
score=0
for i,j in d.items():
    if j>score:
        score=j
        topper=i
print("Topper is",topper,'with score of',score)

```

Output:

Topper is Student1 with score of 54

Q34. Write a program that prints a histogram of frequencies of characters occurring in a message.

Program:

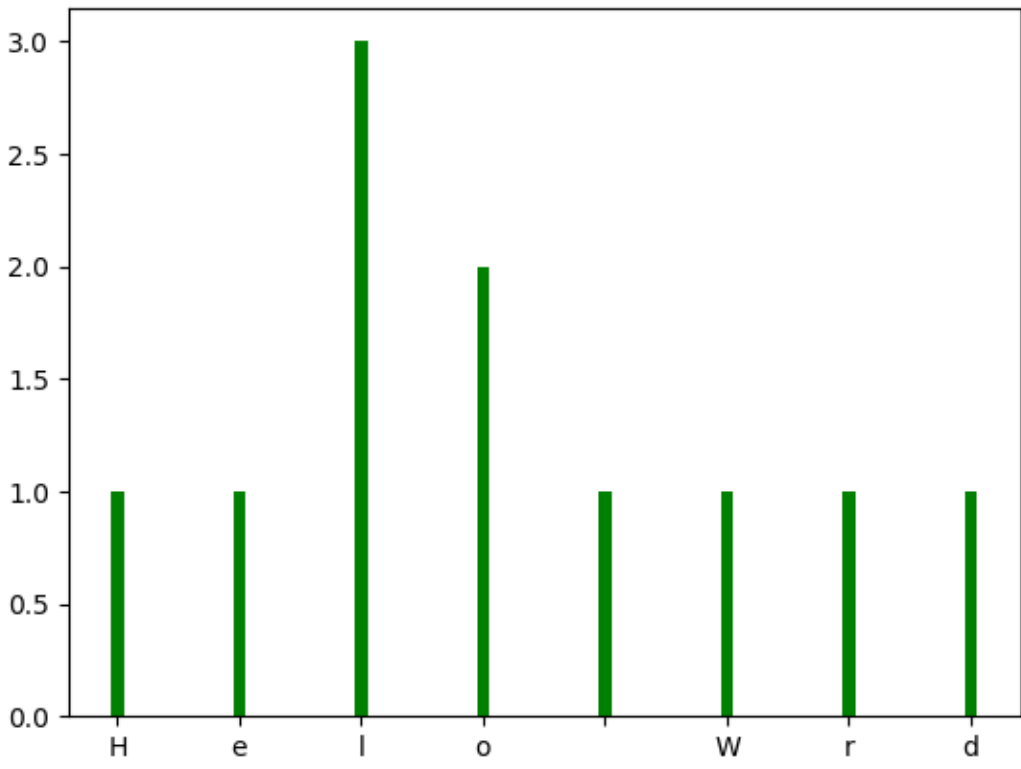
```

a=input("Enter the message: ")
d1={}
for i in a:
    if i in d1:
        d1[i]+=1
    else:
        d1[i]=1
import matplotlib.pyplot as plt
plt.bar(d1.keys(), d1.values(), 0.1, color='g')

```

Output:

Enter the message: Hello World
 <BarContainer object of 8 artists>



Q35. Write a NumPy program to compute the determinant of a given square array.

Program:

```
import numpy as np
import numpy.linalg as npl
a=np.array([[1,2,3],[4,5,6],[3,6,7]])
print("Determinant: ",npl.det(a))
```

Output:

Determinant: 6.000000000000001

Q36. Write a NumPy program to compute the eigenvalues and right eigenvectors of a given square array.

Program:

```
import numpy as np
import numpy.linalg as npl
a=np.array([[1,2,3],[4,5,6],[3,6,7]])
eigenvalues, eigenvectors = npl.eig(a)
print("Eigen Values: ",eigenvalues)
print("Eigen Vectors: ",eigenvectors)
```

Output:

Eigen Values: [13.478192+0.j -0.239096+0.62289379j -0.239096-0.62289379j]
 Eigen Vectors: [[0.2751429 +0.j -0.19948632+0.44791143j -0.19948632-0.44791143j]
 [0.63848541+0.j 0.69533142+0.j 0.69533142-0.j]
 [0.71877171+0.j -0.47416046-0.22642135j -0.47416046+0.22642135j]]

Q37. Write a NumPy program to find a matrix or vector norm.

Program:

```
import numpy as np
import numpy.linalg as npl
a=np.array([[1,2,3],[4,5,6],[3,6,7]])
print("Vector norm:", npl.norm(a))
```

Output:

Vector norm: 13.601470508735444

Q38. Write a NumPy program to compute the inverse of a given matrix.

Program:

```
import numpy as np
import numpy.linalg as npl
a=np.array([[1,2,3],[4,5,6],[3,6,7]])
print("Inverse matrix:", npl.inv(a))
```

Output:

Inverse matrix: [[-1.66666667e-01 6.66666667e-01 -5.00000000e-01]

```
[-1.66666667e+00 -3.33333333e-01 1.00000000e+00]
[ 1.50000000e+00 -2.08166817e-17 -5.00000000e-01]]
```

Date: 29 Sep 2023

Q39 . Read the 'flavors_of_cocoa.csv' file as a dataframe 'df_cocoa' and answer questions that follow:

- Which of the variable have null values?**
- Which of the countries have maximum locations of cocoa manufacturing companies?**
- After checking the data summary, which feature requires a data conversion considering the data values held?**
- What is the maximum rating of chocolates?**

Program:

```
import pandas as pd
df_cocoa=pd.read_csv("/content/flavors_of_cocoa.csv")
print(df_cocoa.isnull().sum())
print()
print("Country name: ",df_cocoa['Company Location'].mode())
print()
print(df_cocoa.info())
print()
print("Maximum Rating: ",df_cocoa['Rating'].max())
```

Ouput:

```
Id          0
Company      0
Bean Origin  0
Review Date  4
Cocoa Percent  0
Company Location  0
Rating       0
dtype: int64
```

```
Country name: 0   U.S.A.
Name: Company Location, dtype: object
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1795 entries, 0 to 1794
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Id              1795 non-null  int64
1   Company         1795 non-null  object
2   Bean Origin     1795 non-null  object
3   Review Date     1791 non-null  float64
4   Cocoa Percent   1795 non-null  object
5   Company Location 1795 non-null  object
6   Rating          1795 non-null  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 98.3+ KB
None
```

```
Maximum Rating: 5.0
```

- a) Review Date
- b) U.S.A
- c) Company, Bean Origin, Cocoa Percent, Company Location
- d) 5.0

Q40. Read the churn.csv dataframe.

- a) What are the total no. of missing values for the variable TotalCharges?
- b) What are the different types of InternetService?
- c) Which is the most popular PaymentMethod?
- d) What is the ratio of the gender?

Program:

```
import pandas as pd
df=pd.read_csv("/content/churn.csv")
print("Total missing values in TotalCharge: ",df['TotalCharges'].isnull().sum())
print("InternetService: ",df['InternetService'].unique())
print("PopularMethod: ",df['PaymentMethod'].mode())
print(df['gender'].value_counts())
```

Output:

```
Total missing values in TotalCharge: 15
InternetService: ['DSL' 'Fiber optic' 'No']
PopularMethod: 0    Electronic check
Name: PaymentMethod, dtype: object
Male          133
Female        124
Name: gender, dtype: int64
```

Q41. Read the Toyota.csv and answer the questions:

- a) Fill in the missing data in the column HP and KM
- b) calculate the range of the KM of the cars
- c) Which field requires a data conversion
- d) Write a code to correct the data values in column Doors.

Program:

```
import pandas as pd
df=pd.read_csv("/content/Toyota.csv")
df['KM'].fillna(df['KM'].mode())
df['HP'].fillna(df['HP'].mode())
df['KM']=df['KM'].replace(to_replace='??',value='0')
print(df['KM'].min())
print(df['KM'].max())
df['Doors']=df['Doors'].replace(to_replace=['three','four','five'],value=['3','4','5'])
```

Output:

```
0
```

99971

KM, FuelType ,HP, Doors require data conversion