① print First 10 natural number using

```
                          # Initialize var to store number
→ num = 1
  while num <= 10;
      print(num)
      num+ = 1
```

② Print the Following pattern

```
1 1 1 1
  1 1 1
    1 1
      1
```

```
→ For i in range(4):
      For j in range(i):
          print(" ", end=" ")
      For k in range(4-i):
          print("1", end=" ")
      Print()
```

③ calculate the sum of all numbers From 1 to given number

```
→  n= int(input("Enter a number:"))
   sum =(n*(n+1))//2
   print(" The sum of all no from 1 to ", n, " is :", sum )
```

④ write a program to print multiplication tabel of given number

```
→  For i in range (1,11):
       print(" 2 x", i, " = ", i*2)
```

⑤ Display numbers From a list using loop

```
→  number = [1,2,3,4,5]
   For num in number
     print(num)
```

⑥ count the total number of digits in ~~um~~ number

```
→  num = input (" Enter a number ")
   num-digit = len (num)
    print(" The total no of digit in no is :", num-digit)
```

⑦ print the Following Pattern : Diamond shape

```
→  height = int(input("Enter height of diamond:" ))
  # For upper part of diamond
      For i in range (1, height+1, 2):
          space = (height-i) //2
      print(" " *space + "*" *i )
```

```
# For lower part of diamond
  For i in range (height -2, 0,-2):
     space =(height -i) //2
     print(" " *space + "*" *i )
```

(8) print list in reverse order using a loop

```
→ list = [1,2,3,4,5]
    For item in reversed(list):
      print(item)
```

(9) Display number from -10 to -1 using For loop

```
→ For num in range (-10,0):
      print(num)
```

(10) use else block to display message " Done" after succesfull execution of For loop

```
→ For num in range (-10,0):
      print(num)
  else:
     print("Done")
```

(11) write a program to display all prime no within a range

```
→ def prime (num):
     if num <= 1:
        return False
     if num <= 3:
        return true
     if num %2 == 0 or num % 3 == 0;
        return False
     i=5
     while i*i <= num:
          if num % i == 0 or num % (i+2)==0:
               return False
          i+=6
     return true
  start = int (input ("Enter starg range :"))
  end = int(input ("Enter end range :"))
```

```
print(F" prime no betn {start} and {end}. ")
    For num in range (start, end+1):
        if prime (num):
            print (num)
```

(12) Display Fibonacci series up to 10 terms

```
→ # Define no of terms you want in fibonacci series
  n= 10
  a,b = 0,1
  print (" Fibonacci series upto 10 terms
    ~~print~~
  For i in range (n):
      print (a, end=" ")
      a,b = b, a+b
  print ()
```

(13) Find Factorial of given number

```
→ n= int (input ("Enter a non-negative no"))
  if n<0:
     print (" Fact is not define for negative no" )
  elif n == 0 :
     print (" Fact of 0 is 1 ")
  else :
     factorial = 1
  # cal factorial using a loop
   for i in range (1, n+1):
       factorial * = i
   print (F" The factorial of {n} is {factorial}" )
```

(14) Reverse a given integer no

```
→ num= int (input ("Enter an integer:"))
  num-str = str (num)
  reversed-str = num-str [::-1]
  reversed-num = int (reversed-str )
  print ("Reversed no:", reversed-num)
```

(10) use a loop to display elem from given list present at odd index position

→
```
list= [1,2,3,4,5,6,7,8,9]
print ("Elem at odd index position :" )
  For i in range(1, len (list),2):
    print (list[i])
```

(11) calculate the cube of all no from 1 to given number

→
```
n=int(input("Enter a number : "))
print ("cubes of no from 1 to", n, ":" )
  For i in range (1, n+1) :
  cube = i**3
  print (F" {i} cube is {cube}" )
```

(12) Find the sum of series upto n terms

→
```
  n= int(input("Enter the no of terms : "))
a = int(input(" Enter the first term : "))
d = int(input ("Enter the common diff :" ))
sum = (n/2) * (2*a + (n-1)*d)
print (F"The sum of series is : {sum}" )
```

(13) print Following pattern

→
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
n= 5
For i in range (1, n+1) :
  for j in range (1 ,i+1 ):
    print (j, end =" " )
  print ()
```

Assignment

Q1 sd = 1·5 $\mu = \frac{2+3+1+3+2+4}{6} = \frac{15}{6} = 2·5$

i $x = 2$

$$z = \frac{x-\mu}{sd} = \frac{2-2·5}{1·5} = \frac{-0·5}{1·5} = -0·333$$

ii $x = 3$

$$z = \frac{x-\mu}{sd} = \frac{3-2·5}{1·5} = \frac{0·5}{1·5} = 0·333$$

iii $x = 1$

$$z = \frac{x-\mu}{sd} = \frac{1-2·5}{1·5} = \frac{-1·5}{1·5} = -1$$

iv $x = 3$

$$z = \frac{x-\mu}{sd} = \frac{3-2·5}{1·5} = \frac{0·5}{1·5} = 0·333$$

v $x = 2$

$$z = \frac{x-\mu}{sd} = \frac{2-2·5}{1·5} = \frac{-0·5}{1·5} = -0·333$$

vi $x = 4$

$$z = \frac{x-\mu}{sd} = \frac{4-2·5}{1·5} = \frac{1·5}{1·5} = 1$$

2) Normalization :

$$x_n = \left[\frac{x - x_{min}}{x_{max} - x_{min}}\right]$$

Q2 One Hot Encoding

One Hot Encoding is a method used to represent categorical variables as binary vector.

Pandas get dummies Function is used for one Hot Encoding

pd.get-dummies(data)

Q3 There are two types of transformer:

① Power Transformer ② Function transformer

① Power transformer

i Box cox

ii Yeo Johnson

② Function Transformer

i log Transformer

ii Reciprocal Transformer

iii Square Transformer

iv square root transformer

v Custom Transformer

Q4 Assumptions of linear Regression -:

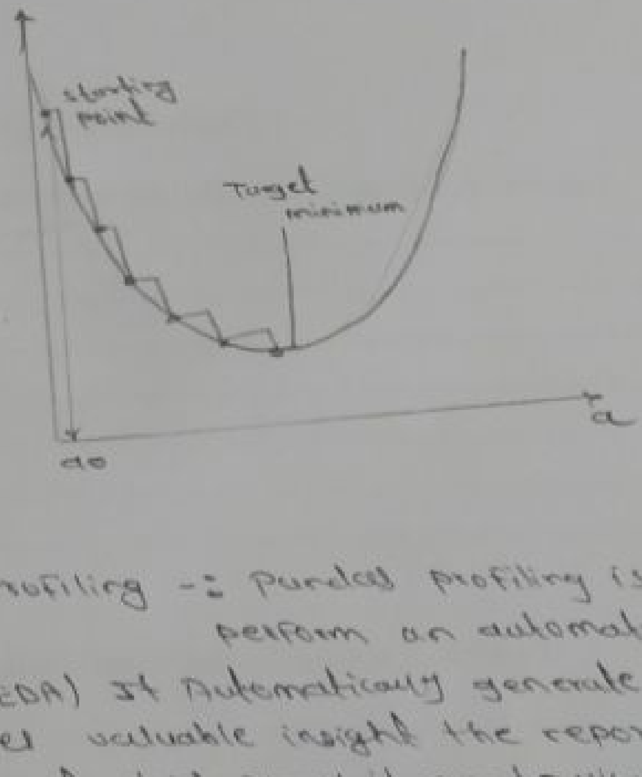
① linearity - It assums that dependent variable (y) changes at constant rate as independent variable changes

② Multivariate Normality - It assums that all the residual value in multiple regression model are normally distributed

③ Homoscedasticity - It means same variance It refer to random noise in relationship betn dependent and independent variable

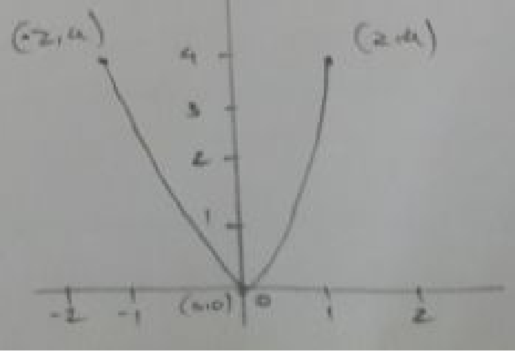
④ Independence - It assums that independent variable in model are not correlated or related to one another

Q5 Gradient Descent Algorithm -: Gradient Descent is an optimization algorithm that is used to train machine learning model. the algorithm minimize the loss function that measure the error betn Predicted values and actual values. the algorithm iteratively adjust the model parameter in the direction of steepest decreases in the loss function.



Q6 Pandas Profiling -: Pandas profiling is python library that perform an automated Exploratory Data Analysis (EDA) It Automatically generate a dataset profile report that gives valuable insight the report is generated in an HTML Format which mean it easy to visualize

Q7



```
q8  Import pandas as pd
    import seaborn as sns
    import matplotlib.pyplot as plt
    From sklearn.model-selection import train-test-split
    From sklearn.linear-model import LinearRegression
ii  df = sns.load-dataset('mpg')
iii (df.isnull().sum() / len(df)) * 100
iv  x = df.Features
    y = df.target
    x-train, xtest, y-train, y-test = train-test-split(x,y, testsize=0.2,
                                  random-state=42)
v   Model = linear Regression()
    Model.Fit(x-train, y-train)
vi  y-pred = Model.predict(x-test)
    y-pred.
```