Q1 —

→ if i=1.
while i<11.
Print (i)
i = i+1

Q2. —

→ n=4
for i in range(4):
for j in range (i+1):
print (" ", end=" ")
for y in range (j,n):
print ("1", end = " ").
Print ( ).

Q3]. —

→ S = int (input())
Sum=0
for i in range (i,s+1):
sum = sum+i.
print (sum).

Q4) → num = int (input ("please enter a Number"))
for i in range (1,11):
Print (num, "*" "=" i num).
print

Q5) — li = [1,2,3,4,5].
for i in (1):
Print (i).

Q6]. Num = int input ().
Total_dig = str (Num).
Print (len (Total_dig)).

Q7] n = int (input ("Enter NO of ROWS")).
for i in range (n):
print (" " * (n-1-1) + "*" * (2*i+1)).
for i in range (n-2, -1, -1)
print (" " * (n-i-1) + "*" (2*i+1)).

Q.8] -
ii = [1,2,3,4,5].
ii. reverse.
ii

using loop -
l = [1,2,3,4,5,6,7].
l = [ ]
for i in st:
i.insert (0, i).
print (i).

Q.9] - n = -10.
for n in range [-10, 0):
print [n].

Q.10] - for i in range(15):
print (i).
else print ("done")

Q11] L = 10
U = 100
print ("prime no. are").
for num in range [1, U+1)
if Num > 1:
for i in range [2, num):
if (num % i) == 0
break.
else:
print (num).

Q.12] —

```
Num = int(input("Enter No"))
n1,n2 = 0,1
count = 0
if nums<=0:
  Print ("please enter a positive number").
else:
  print ("Fibonacci").
  while (ount <nums'
    print (n').
    hth = h1 th2.
    n1 = n2.
    n2 = nth.
    (ount = 1.
```

Q.13] —

→

```
num = int (input ("Enter Number").
fact = 1.
if num < 0:
  Print ("Enter the number").
elif Number == 0:
  print ("The ~~fitc~~torial of 0 is 1").
else.
  for i in range (1, num+1):
    fact = i
  print ("The factorial of "Num, "is"; fact).
```

Q14].

```
Num = int(input("Enter the Number"))
reversed Num = 0
while Num = 0
    digit = num % 10
    reversed_num = reversed_num * 10 + digit
    num // = 10
print("Reversed Number", reversed Num)
```

Q15] –

→

```
my_list = [1,2,3,4,5,6,7]
odd element. my_list [1::2].
print odd_elements).
```

Q16] –

```
Num = int(input("Enter the number"))
for i in range (1, Num+1)
    cube = i * * 3
    Print (f"the cube of {i} is {i**3}")
```

17] –

```
n = input ["Enter the number of terms"]
sum = 0
j = 1
for i in range(1, n+1):
    Sum = sum + j
    j = [j * 10) + 1
```

818) -

```
n = int(input("Enter numberofrows")).
for i in range (1,n+1):
    for j in range (1, i+1):
        print (j, end, " ").
    print ().
```

Assignment-2

Q.1) Calculate the z-score for the below data set assume sd = 1.5 How do u perform normalisation (only formula) Calcu.

2
3
1
3
2
4

→ we have, $z = \frac{x - \mu}{\sigma}$

where, z is the z-score.
x is the individual data point.
μ is the mean.
σ is the standard deviation.

Here $\sigma = 1.5$;
and, the dataset is: 2 3 1 3 2 4
The mean (μ) is cal as.

$$\mu = \frac{2+3+1+2+4}{6} = \frac{15}{16} = 2.5.$$

Now, calculate z-score for each datapoints.

i) for x = 2:

$$z = \frac{2-2.5}{1.5} = \frac{-0.5}{1.5} = -0.333.$$

ii) for x = 3;

$$z = \frac{3-2.5}{1.5} = \frac{0.5}{1.5} = -1.$$

iii) for x = 1:

$$z = \frac{3-2.5}{1.5} = 0.333.$$

iv) for x = 2:

$$z = \frac{2.25}{1.5} = 0.333.$$

vi) for x = 4;

$$z = \frac{4 - 2.5}{1.5} = \frac{1.5}{1.5} = 1.$$

So, the z score for the data set are approximately -0.333, 0.333, -1, 0.333, -0.333, 1.

2].

→ One-hot encoding is a technique used to convert categorical variables into a binary matrix, where each category is represented by a binary column. In pandas, the get_dummies function is commonly used for one last hot encoding.

3) List all the transformers (function and power). The question seems incomplete or ambiguous, if you're referring to mathematical transformations, it could include functions like square root, logarithm, exponential, etc. with different powers.

4) Linear regression assumes that the relationship between the independent variables is linear, and the residuals (the difference between actual and predicted values) are normally distributed with constant variance.

5) Gradient descent is an optimization algorithm used to minimize the cost function in machine learning models. The diagram typically of steepest decrease of the convergence towards the minimum point of cost function.

6) Pandas Profiling is a library for generating exploratory data analysis reports for a pandas Dataframe To use it -

python

copy code -

```
import pandas-profiling.
profile =
pandas — profiling. profile Report (df).
profile. to-file ("output. html").
```

7) Draw the line for the following equation-

$y = x^2$.

=

$2\ y = xc$

2

:

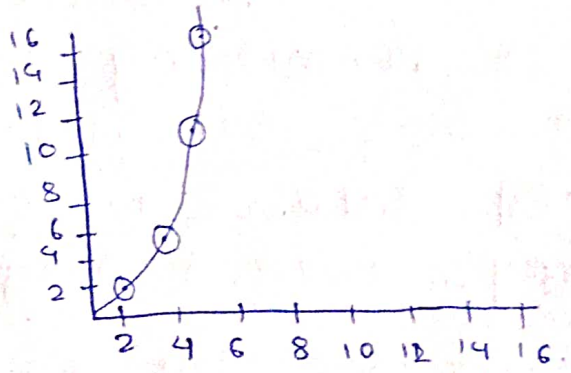
since

=

2.

$y = x$.

2.



represents a quadratic function, the graph is a parabola. it opens upwards if the coefficient of.

2

x

2.

it positive. The specific shape and. location of the parabola. depend on the coefficient values.

8]. python.

copy code.

```
#import neressary libraries.
 import seaborn as sns.
from sklearn.-selection. import.
 train_test_split
 from sklearn. linear.model import.
Linear Regression.
 from sklearn. matics import
  mean-squared-emor.
# Load. the dataset.
 mpg- data= sns.load=dataset ("mpg")
# check for missing values.
  print (mpg_ data.isnull().sum()),
x =mpg_data. [[ 'horse power']].
y= mpg_ data. ["mpg"].

 x = train ;xc= test , y - train , y - test=
  train = test_split (x, y ;test_size =0,2).
  random_state = 42).
 # create and fill the model.
    model = Linear Regression.

  # Evaluate the model.
 mse = mean_sequenced_error (y-test, y-pre
 print(f' mean squared Error : {mse}').
```

This code imports the necessary libaries, loads the "mpg" dataset, checks for the missing values. splits data into training and testing datasets, biulds a linear regression model. makes predications, and evaluates the model. using mean squared error.