**Name :- Shivam Indrabhan Borse**

**Roll No :- 19**

**Subject: Artificial Neural Network**

**Class : TE**

**Branch: AI & DS**

**Practical – 3**

**Problem statement :** Write a Python Program using Perceptron Neural Network to recognise even and odd numbers. Given numbers are in ASCII form 0 to 9

**Code :**

import numpy as np

class Perceptron:

    def \_\_init\_\_(self, input\_size, learning\_rate=0.1):

        self.weights = np.zeros(input\_size)

        self.bias = 0

        self.learning\_rate = learning\_rate

    def predict(self, inputs):

        weighted\_sum = np.dot(self.weights, inputs) + self.bias

        if weighted\_sum >= 0:

            return 1

        else:

            return 0

    def train(self, inputs, label):

        prediction = self.predict(inputs)

        error = label - prediction

        self.weights += self.learning\_rate \* error \* inputs

        self.bias += self.learning\_rate \* error

training\_data = {

    "48": 1,

    "49": 0,

    "50": 1,

    "51": 0,

    "52": 1,

    "53": 0,

    "54": 1,

    "55": 0,

    "56": 1,

    "57": 0,

}

perceptron = Perceptron(input\_size=7)

for ascii\_code, label in training\_data.items():

    inputs = np.array([int(b) for b in bin(int(ascii\_code))[2:].zfill(7)])

    perceptron.train(inputs, label)

test\_data = {

    "48": "even",

    "49": "odd",

    "50": "even",

    "51": "odd",

    "52": "even",

    "53": "odd",

    "54": "even",

    "55": "odd",

    "56": "even",

    "57": "odd",

}

for ascii\_code, expected\_output in test\_data.items():

    inputs = np.array([int(b) for b in bin(int(ascii\_code))[2:].zfill(7)])

    output = perceptron.predict(inputs)

    if output == 1:

        print(f"'{chr(int(ascii\_code))}' is {expected\_output}.")

    else:

        print(f"'{chr(int(ascii\_code))}' is {expected\_output}.")

**Output :**

