**Logo

Description automatically generated**

**San Francisco Bay University**

**CS483L - Artificial Intelligence & Machine Learning Lab**

**2022 Summer Midterm Exam**

**Student Name: Mahmud Omer Student ID:19660**

**Instruction**

1. **Put your answer right after each question in the answer sheet**
2. **Make copy & paste for your program in text mode, NOT image except running results onto the answer sheet.**

**Part I Write python programs by the functions in *Numpy***

1. Create 3x3 matrix with random number elements

**Solution:**

|  |
| --- |
| import numpy as np  randMat = np.random.randint(0,5,(3,3))  randMat |

1. Don’t change original array, substitute even numbers in it to -1, such as

*a = [11, 12, 13, 14]*

*b = [11, -1, 13, -1]*

**Solution:**

|  |
| --- |
| arr = [11, 12, 13, 14]  for i in range(len(arr)):    if arr[i]%2 == 0:      arr[i] = -1  arr |

1. Extract the common elements from two arrays, like

*a = [11,12, 13, 14] and b = [11,13, 15, 16]*

*c = [11, 13]*

**Solution:**

|  |
| --- |
| a = [11,12, 13, 14]  b = [11,13, 15, 16]  c = []  for i in range(len(a)):    if len(a)<=len(b):      for j in range(len(b)):        if a[i] == b[j]:          c.append(a[i])          continue    else:       for j in range(len(a)):        if b[i] == c[j]:          c.append(a[i])          continue  print(c) |

1. Swap 2 rows in a 3x3 matrix

**Solution:**

|  |
| --- |
| # 4.  Swap 2 rows in a 3x3 matrix  a = np.array([[1,2,3], [3,4,5], [5,6,7]])  print(a)  # swap two raws ( lets swap 1st and 2nd)  a[[0,1]] = a[[1, 0]]  print(a) |

1. Calculate mean and standard deviation in an integer array

**Solution:**

|  |
| --- |
| # 5. Calculate mean and standard deviation in an integer array  a = np.array([1, 2, 3, 4, 5, 6, 7])  print(a)    mean = np.mean(a)  print("\nMean: ", mean)    standarDeviation = np.std(a)  print("\nstandard Deviation: ", standarDeviation) |

1. Convert string 'P' and 'NP' to number 0 and 1 respectively in a given 1d array, such as

*a = [1, 2, 3, 'P', 4, 5, 6, 'NP']*

*b = [1, 2, 3, 0, 4, 5, 6, 1]*

**Solution:**

|  |
| --- |
| #6. Convert string 'P' and 'NP' to number 0 and 1 respectively in a given 1d array, such as  # a = [1, 2, 3, 'P', 4, 5, 6, 'NP']  # b = [1, 2, 3, 0, 4, 5, 6, 1]  a = [1, 2, 3, 'P', 4, 5, 6, 'NP']  b = []  for i in range(len(a)):    if a[i] == 'P':      b.append(0)    elif a[i] == "NP":        b.append(1)    else:        b.append(a[i])  print("a = ",a)  print("b = ", b) |

**Part II Complete the following questions in python programs by the functions in *Matplotlib***

1. Plot two lines on one figure with the legend for each

**Solution:**

|  |
| --- |
| #7. Plot two lines on one figure with the legend for each  import matplotlib.pyplot as plt  x= [2, 3, 4, 5, 6, 7, 7, 8]  y1 = [3, 4, 3,6, 7, 5, 6, 8]  y2 = [4, 5, 4, 7, 8, 6 , 7, 9]  plt. plot(x,y1, label='y1 Graph')  plt.plot(x,y2, label=' y2 Graph')  plt.xlabel('x')  plt.ylabel('y')  plt.title('random graph')  plt.legend()  plt.show() |

1. Draw a 2d scatter plot for samples (X1, X2, y) in different colors for y = 0 and y = 1

**Solution:**

|  |
| --- |
| #8. Draw a 2d scatter plot for samples (X1, X2, y) in different colors for y = 0 and y = 1  import matplotlib.pyplot as plt  x1= [2, 3, 4, 5, 6, 7, 7, 8]  x2 = [3, 4, 3,6, 7, 5, 6, 8]  y = [0, 1, 1, 0, 0, 1 , 1, 0]  plt.scatter(x1, x2 ,c= y)  plt.xlabel('x1')  plt.ylabel('x2')  plt.title('random graph')  plt.legend()  plt.show() |

**Part III Create python program from *Pandas* modules**

1. Change 1d array from 0 to 8 to 3x3 dataframe(matrix)

**Solution:**

|  |
| --- |
| #9. Change 1d array from 0 to 8 to 3x3 dataframe(matrix)  import numpy as np  import pandas as pd  ar1= np.array([0,1,2,3,4,5,6,7,8])  ar2 = np.reshape(ar1, (-1, 3))  numpyData = np.array(ar2)  data\_frame = pd.DataFrame(data=numpyData, index=["r1", "r2","r3"], columns=["c1", "c2","c3"])  print(data\_frame) |

1. Calculate the Euclidean distance from two 1d arrays, such as

a = [1, 2, 3] and b = [4, 5, 6]

**Solution:**

|  |
| --- |
| # 10. Calculate the Euclidean distance from two 1d arrays, such as  # a = [1, 2, 3] and b = [4, 5, 6]  from numpy.ma.core import sqrt  a = [1, 2, 3]  b = [4, 5, 6]  ec\_d = 0  for i in range(len(a)):     d1 = a[i]     d2 = b[i]     ec\_d += (d1- d2)\*\*2  ec\_d = sqrt(ec\_d)  print("Eucludian Distance =", ec\_d) |

1. Only import odd number rows from a csv file

**Solution:**

|  |
| --- |
| # 11. Only import odd number rows from a csv file  import numpy as np  import pandas as pd  # loading file from google collab  from google.colab import drive  drive.mount('/content/drive')  path = ('/content/q11.txt')  # file path  data = pd.read\_csv(path) # load data  data = pd.DataFrame(data[::2]) # read every next line which is 1, 3, 5 ..... etc |

1. Substitute all elements in diagonals to 0 in a 5x5 dataframe(matrix)

**Solution:**

|  |
| --- |
| #12.  Substitute all elements in diagonals to 0 in a 5x5 dataframe(matrix)  import numpy as np  import pandas as pd  dataFrame = pd.DataFrame(np.random.rand(5,5))  dataFrame.values[[np.arange(dataFrame.shape[0])]\*2] = 0  print(dataFrame) |

1. Separate a dataframe 80% to trainingset and 20% as testset

**Solution:**

|  |
| --- |
| #13.  Separate a dataframe 80% to trainingset and 20% as testset  from sklearn.model\_selection import train\_test\_split  import numpy as np  import pandas as pd  # using the random data from 12  dataFrame = pd.DataFrame(np.random.rand(5,5))  dataFrame.values[[np.arange(dataFrame.shape[0])]\*2] = 0  # dividing data in two x and y  X = dataFrame.iloc[:, :-1].values         # Assign 1st/2nd/3rd/4th colums values to X  y = dataFrame.iloc[:, 4].values           # Assign 5th column values to y  # dividing the data in to training and testing sets  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)  # dataframe divided in to 80% training set and 20% as test set |