In [1]:
```
#Practical 1
```

In [2]:
```
num1=int(input("Enter your 1st number:"))
num2=int(input("Enter your 2nd number:"))
total=num1+num2
print("Addition of num1 and num2 is ",total)
```

```
Enter your 1st number:3
Enter your 2nd number:3
Addition of num1 and num2 is  6
```

In [3]:
```
#Practical 2
```

In [8]:
```
num=int(input("Enter the number:"))
factorial=1

if num==0 or num==1:
    print("Factorial of 0 and 1 is 1.")

elif num<0:
    print("Factorial of negative number is does't exist.")

else:
    for i in range(1,num+1):
        factorial=factorial*i
    print("Factorial of",num,"is",factorial)
```

```
Enter the number:5
Factorial of 5 is 120
```

In [9]:
```
#Practical 3
```

In [14]:
```
num=int(input("Enter the number:"))
print("here is the table of",num)
for i in range(1,11):
    table=num*i
    print(num,'x',i,'=',table)
```

```
Enter the number:2
Here is the table of 2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

In [15]:
```
#Practical 4
```

In [17]:
```
num=int(input("Enter the number:"))
if num%2==0:
    print(num,"is even number.")
else:
    print(num,"is odd number.")
```

```
Enter the number:5
5 is odd number.
```

In [18]:
```
#Practical 5
```

In [20]:
```
import numpy as np
arr=np.array([1,2,3,4,5])
print(arr)
print(arr.ndim)
```

```
[1 2 3 4 5]
1
```

In [21]:
```
#Practical 6
```

In [27]:
```
import numpy as np
arr1=np.array([1,2,3,4,5,6,7,8])
print(arr1[2:5])

print(arr1[3:])
```

```
[3 4 5]

[4 5 6 7 8]
```

In [28]:
```
#Practical 7
```

In [32]:
```
from scipy import constants
print(constants.pi)
print(constants.mega)
print(constants.kilo)
```

```
3.141592653589793
1000000.0
1000.0
```

In [33]:
```
#Practical 8
```

In [34]:
```
from scipy.optimize import root
from math import cos

def eqn(x):
    return x+cos(x)
my_root=root(eqn,0)
print(my_root.x)
```

```
[-0.73908513]
```

In [35]:
```
#Practical 9
```

In [39]:
```
import pandas as pd
data={"Names":["ABC","PQR","XYZ"],
      "Marks":[75,60,95]}
df=pd.DataFrame(data)
print(df)
```

```
  Names  Marks
0   ABC     75
1   PQR     60
2   XYZ     95
```

In [40]:
```
#Practical 10
```

In [41]:
```
import pandas as pd
data=[23,34,56]
df=pd.Series(data)
print(df)
```

```
0    23
1    34
2    56
dtype: int64
```

In [42]:
```
#Practical 11
```

In [44]:
```
from sklearn.datasets import load_iris
iris=load_iris()
x=iris.data
y=iris.target
feature_name=iris.feature_names
target_name=iris.target_names
print("Feature names:",feature_name)
print("Target names:",target_name)
print("First 5 rows of x:\n",x[:5])
```

```
Feature names: ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
Target names: ['setosa' 'versicolor' 'virginica']
First 5 rows of x:
 [[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]]
```

In [45]:
```
#Practical 12
```

In [49]:
```
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
iris=load_iris()
x=iris.data
y=iris.target

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4,random_state=1)
classifier_knn=KNeighborsClassifier(n_neighbors=3)
classifier_knn.fit(x_train,y_train)
y_pred=classifier_knn.predict(x_test)

print("Accuracy:",metrics.accuracy_score(y_test,y_pred))
sample=[[5,5,3,2],[2,3,4,5]]
preds=classifier_knn.predict(sample)
pred_species=[iris.target_names[i] for i in preds]
print("Predictions:",pred_species)
```

```
Accuracy: 0.9833333333333333
Predictions: ['versicolor', 'virginica']
```

In [50]:
```
#Practical 13
```

In [54]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.model_selection import train_test_split
import numpy
# fix random seed for reproducibility
seed=7
numpy.random.seed(seed)
# load pima indians dataset
dataset = numpy.loadtxt("pima-indians-diabetes.csv", delimiter=",")
# split into input (X) and output (Y) variables
X = dataset[:,0:8]
Y = dataset[:,8]
# split into 67% for train and 33% for test
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.33,
random_state=seed)
# create model
model = Sequential()
model.add(Dense(12, input_dim=8, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
# Compile model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# fit the model
model.fit(X_train, y_train, validation_data=(X_test,y_test), epochs=150, batch_size=10)
```

In [55]:
```
#Practical 14
```

In [57]:
```
from nltk.tokenize import sent_tokenize,word_tokenize
text="Hello! This is a nltk program"
print(sent_tokenize(text))
print(word_tokenize(text))
```

```
['Hello!', 'This is a nltk program']
['Hello', '!', 'This', 'is', 'a', 'nltk', 'program']
```

In [58]:
```
#Practical 15
```

In [63]:
```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
stop_words=set(stopwords.words("english"))
data=input("Enter your sentence:")
words=word_tokenize(data)
filter_words=[]

for i in words:
    if i not in stop_words:
        filter_words.append(i)
print(filter_words)
```

```
Enter your sentence:This a python programmer and data science student.
['This', 'python', 'programmer', 'data', 'science', 'student', '.']
```

In [64]:
```
#Practical 16
```

In [66]:
```
from nltk.stem import WordNetLemmatizer
lemmatizer=WordNetLemmatizer()
lemmatizer.lemmatize('cars')
```

Out[66]:
```
'car'
```

In [67]:
```
#Practical 17
```

In [70]:
```
from nltk.stem import PorterStemmer
ps=PorterStemmer()
words=['game','games','gaming']
for i in words:
    print(i,":",ps.stem(i))
```

```
game : game
games : game
gaming : game
```

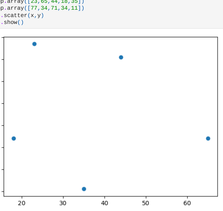
In [71]:
```
#Practical 18
```

In [72]:
```
from nltk.tokenize import word_tokenize
sentence="This is a Program"
print(nltk.pos_tag(word_tokenize(sentence)))
```

```
[('This', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('Program', 'NN')]
```

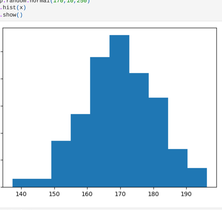
In [73]:
```
#Practical 19
```

In [77]:
```
import matplotlib.pyplot as plt
import numpy as np
x=np.array([23,65,44,18,35])
y=np.array([77,34,71,34,11])
plt.scatter(x,y)
plt.show()
```



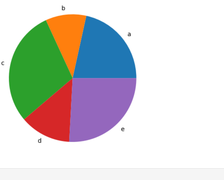
In [78]:
```
#Practical 20
```

In [84]:
```
import matplotlib.pyplot as plt
import numpy as np
x=np.random.normal(170,10,250)
plt.hist(x)
plt.show()
```



In [85]:
```
#Practical 21
```

In [87]:
```
import matplotlib.pyplot as plt
import numpy as np
x=np.array([25,12,34,15,30])
plt.pie(x,labels=['a','b','c','d','e'])
plt.show()
```



In [ ]: