

Python Project demo file

12 science batch-2020-21

Daksh Prajapati

Darshan Dholariya

Maharshi Pipaliya





```

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import sys

from PIL import Image


#Title of project
cstr="POKEMON"

print(cstr.center(80,"#"))

print("  ")


#pokemon info

print("The original Pokémon is a role-playing game based around building a small team of monsters to battle other monsters in a quest to become the best. Pokémon are divided into types, such as water and fire, each with different strengths. Battles between them can be likened to the simple hand game rock-paper-scissors. For example, to gain an advantage over a Pokémon that cannot beat an opponent's Charizard character because of a weakness to fire, a player might substitute a water-based Pokémon. With experience, Pokémon grow stronger, gaining new abilities. By defeating Gym Leaders and obtaining Gym Badges, trainers garner acclaim.")

print("  ")


#####      graphs      #####

ax=plt.axes()


def Bugbar():

    Bug=pd.read_csv("D:\python projects\schoolproject\Bug.csv")

    Namei=Bug["Name"]

    HPi=Bug["HP"]

    Attacki=Bug["Attack"]

    Defensei=Bug["Defense"]

    plt.xlabel("Name of pokemon")

    ax=plt.axes()

```



```
print("press 1 for the data for name of pokemon Vs HP")
print("press 2 for the data for name of pokemon Vs Attack")
print("press 3 for the data for name of pokemon Vs Defense")
print("press 4 to merge all the data in one stackbar chart")
print("press 5 to merge all the data in one multibar chart")
```

```
pt=int(input("enter what type of graph u want to see:"))
```

```
if pt==1:
```

```
    bugbar=ax.set_facecolor("navajowhite")
    plt.ylabel("HP of pokemons")
    plt.title("Name Vs HP")
    plt.bar(Namei,HPi)
    plt.xticks(rotation="vertical")
    plt.show()
```

```
elif pt==2:
```

```
    bugbar=ax.set_facecolor("bisque")
    plt.ylabel("attaking strength of pokemons")
    plt.title("Name Vs Attack strength")
    plt.bar(Namei,Attacki,color="r")
    plt.xticks(rotation="vertical")
    plt.show()
```

```
elif pt==3:
```

```
    bugbar=ax.set_facecolor("lightpink")
    plt.ylabel("defense strength of pokemons")
    plt.title("Name Vs defense strength")
    plt.bar(Namei,Defensei,color="m")
    plt.xticks(rotation="vertical")
```



```
plt.show()
```

```
elif pt==4:
```

```
    bugbar=ax.set_facecolor("peachpuff")
```

```
    plt.ylabel("HP,Attack and defense strength")
```

```
    plt.title("Name Vs all abilities")
```

```
    plt.bar(Namei,HPi,width=0.2,label="Name Vs HP")
```

```
    plt.bar(Namei,Attacki,width=0.2,label="Name Vs Attack strength")
```

```
    plt.bar(Namei,Defensei,width=0.2,label="Name Vs defense strength")
```

```
    plt.xticks(rotation="vertical")
```

```
    plt.legend()
```

```
    plt.show()
```

```
elif pt==5:
```

```
    bugbar=ax.set_facecolor("peachpuff")
```

```
    VD=np.arange(len(Namei))
```

```
    plt.bar(VD,HPi,width=0.33,label="Name Vs HP")
```

```
    plt.bar(VD+0.33,Attacki,width=0.33,label="Name Vs Attack strength")
```

```
    plt.bar(VD+0.66,Defensei,width=0.33,label="Name Vs defense strength")
```

```
    plt.xticks(VD,Namei,rotation="vertical")
```

```
    plt.title("Name Vs all abilities")
```

```
    plt.legend()
```

```
    plt.show()
```

```
else:
```

```
    print("enter valid input")
```

```
def Bughistogram():
```

```
    ax=plt.axes()
```

```
    Bug=pd.read_csv("D:\python  
projects\schoolproject\Bug.csv")
```



```

Totali=Bug["Total"]
Bughisto=ax.set_facecolor("lightgoldenrodyellow")
plt.hist(Totali,bins=[0,100,200,300,400,500,634],color="g")
plt.ylabel("Total Number of different BUG pokemons")
plt.title("Range wise total number of bug pokemons")
plt.xticks(rotation="vertical")
plt.show()

```

```

def Bugline():
    Bug=pd.read_csv("D:\python projects\schoolproject\Bug.csv")
    Namei=Bug["Name"]
    HPi=Bug["HP"]
    Attacki=Bug["Attack"]
    Defensei=Bug["Defense"]
    plt.xlabel("Name of pokemon")
    plt.xticks(rotation="vertical")
    ax=plt.axes()

    print("press 1 to print the data for POKEMON v/s HP")
    print("press 2 to print the data for POKEMON v/s ATTACK")
    print("press 3 to print the data for POKEMON v/s DEFENCE")
    print("press 4 to print the data for POKEMON v/s ALL ABILITIES")

    pt=int(input("Enter your choice="))

    if pt==1:
        bugline=ax.set_facecolor("steelblue")
        plt.ylabel("HP of pokemons")
        plt.title("Name Vs HP")

```



```

plt.plot(Namei,HPi,color="w",marker='o',markersize=8,markeredgecolor='midnightblue',ma
rkerfacecolor="lavenderblush",linestyle="dashed",linewidth=2,label="NAME V/S HP")

plt.show()

elif pt==2:

    bugline=ax.set_facecolor("steelblue")

    plt.ylabel("attaking strength of pokemons")

    plt.title("Name Vs Attack strength")

plt.plot(Namei,Attacki,color="w",marker='o',markersize=8,markeredgecolor='midnightblue
',markerfacecolor="lavenderblush",linestyle="dashed",linewidth=2,label="NAME V/S HP")

plt.show()

elif pt==3:

    bugline=ax.set_facecolor("steelblue")

    plt.ylabel("defense strength of pokemons")

    plt.title("Name Vs defense strength")

plt.plot(Namei,Defensei,color="w",marker='o',markersize=8,markeredgecolor='midnightblu
e',markerfacecolor="lavenderblush",linestyle="dashed",linewidth=2,label="NAME V/S HP")

plt.show()

elif pt==4:

    bugline=ax.set_facecolor("antiquewhite")

    plt.ylabel("HP,Attack and defense strength")

    plt.title("Name Vs abilities")

    plt.plot(Namei,HPi,color="r",label="NAME V/S HP")

    plt.plot(Namei,Attacki,color="c",label="NAME V/S ATTACK")

    plt.plot(Namei,Defensei,color="m",label="NAME      V/S
DEFENSE")

    plt.legend()

    plt.show()

```




```

else:
    print("enter valid input")

def Bugpie():
    Bug=pd.read_csv("D:\python projects\schoolproject\Bug.csv")
    Namei=Bug["Name"]
    HPi=Bug["HP"]
    Attacki=Bug["Attack"]
    Defensei=Bug["Defense"]

    print("press 1 for the data for name of pokemon Vs HP of pokemons")
    print("press 2 for the data for name of pokemon Vs Attack strength of pokemons")
    print("press 3 for the data for name of pokemon Vs Defense strength of pokemons")

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:
        plt.title("name vs HP")
        plt.pie(HPi,labels=Namei)
        #autopct attribute can be used to show the percentage of the data.
        plt.show()
    elif pt==2:
        plt.title("Name Vs Attack strength")
        plt.pie(Attacki,labels=Namei)
        plt.show()

    elif pt==3:
        plt.title("Name Vs defense strength")
        plt.pie(Defensei,labels=Namei)
        plt.show()
    else:

```



```

    print("enter void input")

def Bugscatter():
    Bug=pd.read_csv("D:\python projects\schoolproject\Bug.csv")
    Namei=Bug["Name"]
    HPi=Bug["HP"]
    Attacki=Bug["Attack"]
    Defensei=Bug["Defense"]
    plt.xlabel("Name of pokemons")
    plt.xticks(rotation="vertical")

    print("press 1 for the data for name of pokemon Vs HP")
    print("press 2 for the data for name of pokemon Vs Attack")
    print("press 3 for the data for name of pokemon Vs Defense")
    print("press 4 to merge all the data in one scatter chart")

    ax=plt.axes()
    axs=plt.gca()

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:
        bugscatter=ax.set_facecolor("lightsteelblue")
        plt.ylabel("HP of pokemons")
        axs.scatter(Namei,HPi,color="k")
        plt.title("Name vs HP")
        plt.show()

    elif pt==2:
        bugscatter=ax.set_facecolor("lightsteelblue")
        plt.ylabel("Attack strength of pokemons")

```



```
    axs.scatter(Namei,Attacki,color="royalblue")  
    plt.title("Name vs Attack strength")  
    plt.show()
```

```
elif pt==3:
```

```
    bugscatter=ax.set_facecolor("lightsteelblue")  
    plt.ylabel("Defense strength of pokemons")  
    axs.scatter(Namei,Defensei,color="navy")  
    plt.title("Name Vs defense strength")  
    plt.show()
```

```
elif pt==4:
```

```
    bugscatter=ax.set_facecolor("lightsteelblue")  
    plt.ylabel("All abilities of pokemons")  
    axs.scatter(Namei,HPi,color="k",label="Name Vs HP")  
    axs.scatter(Namei,Attacki,color="royalblue",label="Name Vs Attack")  
    axs.scatter(Namei,Defensei,color="navy",label="Name Vs Defense")  
    plt.title("complete analysis of pokemons")  
    plt.legend()  
    plt.show()
```

```
else:
```

```
    print("enter valid input")
```

```
def Darkbar():
```

```
    Dark=pd.read_csv("D:\python projects\schoolproject\Bug.csv")  
    Namei=Dark["Name"]  
    HPi=Dark["HP"]  
    Attacki=Dark["Attack"]  
    Defensei=Dark["Defense"]  
    plt.xlabel("Name of pokemon")
```



```

ax=plt.axes()

print("press 1 for the data for name of pokemon Vs HP")
print("press 2 for the data for name of pokemon Vs Attack")
print("press 3 for the data for name of pokemon Vs Defense")
print("press 4 to merge all the data in one stackbar chart")
print("press 5 to merge all the data in one multibar chart")

pt=int(input("enter what type of graph u want to see:"))

if pt==1:
    darkbar=ax.set_facecolor("navajowhite")
    plt.ylabel("HP of pokemons")
    plt.title("Name Vs HP")
    plt.bar(Namei,HPi)
    plt.xticks(rotation="vertical")
    plt.show()

elif pt==2:
    darkbar=ax.set_facecolor("bisque")
    plt.ylabel("attaking strength of pokemons")
    plt.title("Name Vs Attack strength")
    plt.bar(Namei,Attacki,color="r")
    plt.xticks(rotation="vertical")
    plt.show()

elif pt==3:
    darkbar=ax.set_facecolor("lightpink")
    plt.ylabel("defense strength of pokemons")
    plt.title("Name Vs defense strength")
    plt.bar(Namei,Defensei,color="m")

```



```
plt.xticks(rotation="vertical")
```

```
plt.show()
```

```
elif pt==4:
```

```
    darkbar=ax.set_facecolor("peachpuff")
```

```
    plt.ylabel("HP,Attack and defense strength")
```

```
    plt.title("Name Vs all abilities")
```

```
    plt.bar(Namei,HPi,width=0.2,label="Name Vs HP")
```

```
    plt.bar(Namei,Attacki,width=0.2,label="Name Vs Attack strength")
```

```
    plt.bar(Namei,Defensei,width=0.2,label="Name Vs defense strength")
```

```
    plt.xticks(rotation="vertical")
```

```
    plt.legend()
```

```
    plt.show()
```

```
elif pt==5:
```

```
    darkbar=ax.set_facecolor("peachpuff")
```

```
    VD=np.arange(len(Namei))
```

```
    plt.bar(VD,HPi,width=0.33,label="Name Vs HP")
```

```
    plt.bar(VD+0.33,Attacki,width=0.33,label="Name Vs Attack strength")
```

```
    plt.bar(VD+0.66,Defensei,width=0.33,label="Name Vs defense strength")
```

```
    plt.xticks(VD,Namei,rotation="vertical")
```

```
    plt.title("Name Vs all abilities")
```

```
    plt.legend()
```

```
    plt.show()
```

```
else:
```

```
    print("enter valid input")
```

```
def Darkhistogram():
```

```
    Dark=pd.read_csv("D:\python  
projects\schoolproject\Bug.csv")
```



```

Totali=Dark["Total"]
ax=plt.axes()
Darkhisto=ax.set_facecolor("paleturquoise")
plt.hist(Totali,bins=[300,400,500,600],color="k")
plt.ylabel("Total Number of different DARK pokemons")
plt.title("Range wise total number of Dark pokemons")
plt.xticks(rotation="vertical")
plt.show()

```

```

def Darkline():
    Dark=pd.read_csv("D:\python projects\schoolproject\Dark.csv")
    Namei=Dark["Name"]
    HPi=Dark["HP"]
    Attacki=Dark["Attack"]
    Defensei=Dark["Defense"]
    plt.xlabel("Name of pokemon")
    plt.xticks(rotation="vertical")
    ax=plt.axes()

    print("press 1 to print the data for POKEMON v/s HP")
    print("press 2 to print the data for POKEMON v/s ATTACK")
    print("press 3 to print the data for POKEMON v/s DEFENCE")
    print("press 4 to print the data for POKEMON v/s ALL ABILITIES")

    pt=int(input("Enter your choice="))

    if pt==1:
        Darkline=ax.set_facecolor("lightsteelblue")
        plt.ylabel("HP of pokemons")
        plt.title("Name Vs HP")

```



```

plt.plot(Namei,HPi,color="k",marker='p',markersize=10,markeredgecolor='c',markerfacecolor="darkslategrey",linestyle="dashdot",linewidth=2,label="NAME V/S HP")

plt.show()

elif pt==2:

    Darkline=ax.set_facecolor("lightsteelblue")

    plt.ylabel("attacking strength of pokemons")

    plt.title("Name Vs Attack strength")

plt.plot(Namei,Attacki,color="k",marker="p",markersize=10,markeredgecolor='c',markerfacecolor="darkslategrey",linestyle="dashdot",linewidth=2,label="NAME V/S ATTACK")

plt.show()

elif pt==3:

    Darkline=ax.set_facecolor("lightsteelblue")

    plt.ylabel("defense strength of pokemons")

    plt.title("Name Vs defense strength")

plt.plot(Namei,Defensei,color="k",marker="p",markersize=10,markeredgecolor='c',markerfacecolor="darkslategrey",linestyle="dashdot",linewidth=2,label="NAME V/S DEFENSE")

plt.show()

elif pt==4:

    Darkline=ax.set_facecolor("linen")

    plt.ylabel("HP,Attack and defense strength")

    plt.title("Name Vs abilities")

    plt.plot(Namei,HPi,color="b",label="NAME V/S HP")

    plt.plot(Namei,Attacki,color="g",label="NAME V/S ATTACK")

    plt.plot(Namei,Defensei,color="m",label="NAME V/S DEFENSE")

    plt.legend()

    plt.show()

```



```

else:
    print("enter valid input")

def Darkpie():
    Dark=pd.read_csv("D:\python projects\schoolproject\Dark.csv")
    Namei=Dark["Name"]
    HPi=Dark["HP"]
    Attacki=Dark["Attack"]
    Defensei=Dark["Defense"]

    print("press 1 for the data for name of pokemon Vs HP of pokemons")
    print("press 2 for the data for name of pokemon Vs Attack strength of pokemons")
    print("press 3 for the data for name of pokemon Vs Defense strength of pokemons")

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:
        plt.title("name vs HP")
        plt.pie(HPi,labels=Namei)
        #autopct attribute can be used to show the percentage of the data.
        plt.show()
    elif pt==2:
        plt.title("Name Vs Attack strength")
        plt.pie(Attacki,labels=Namei)
        plt.show()
    elif pt==3:
        plt.title("Name Vs defense strength")
        plt.pie(Defensei,labels=Namei)
        plt.show()
    else:
        print("enter vaid input")

```




```

def Darkscatter():
    Dark=pd.read_csv("D:\python projects\schoolproject\Dark.csv")
    Namei=Dark["Name"]
    HPi=Dark["HP"]
    Attacki=Dark["Attack"]
    Defensei=Dark["Defense"]
    plt.xlabel("Name of pokemons")
    plt.xticks(rotation="vertical")

    print("press 1 for the data for name of pokemon Vs HP")
    print("press 2 for the data for name of pokemon Vs Attack")
    print("press 3 for the data for name of pokemon Vs Defense")
    print("press 4 to merge all the data in one scatter chart")

    ax=plt.axes()
    axs=plt.gca()

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:
        Darkscatter=ax.set_facecolor("lightsteelblue")
        plt.ylabel("HP of pokemons")
        axs.scatter(Namei,HPi,color="brown")
        plt.title("Name vs HP")
        plt.show()

    elif pt==2:
        Darkscatter=ax.set_facecolor("lightsteelblue")
        plt.ylabel("Attack strength of pokemons")
        axs.scatter(Namei,Attacki,color="k")

```



```
plt.title("Name vs Attack strength")
```

```
plt.show()
```

```
elif pt==3:
```

```
    Darkscatter=ax.set_facecolor("lightsteelblue")
```

```
    plt.ylabel("Defense strength of pokemons")
```

```
    axs.scatter(Namei,Defensei,color="navy")
```

```
    plt.title("Name Vs defense strength")
```

```
    plt.show()
```

```
elif pt==4:
```

```
    Darkscatter=ax.set_facecolor("lightgoldenrodyellow")
```

```
    plt.ylabel("ALL abilities of pokemons")
```

```
    axs.scatter(Namei,HPi,color="r",label="Name Vs HP")
```

```
    axs.scatter(Namei,Attacki,color="g",label="Name Vs Attack")
```

```
    axs.scatter(Namei,Defensei,color="m",label="Name Vs Defense")
```

```
    plt.title("complete analysis of pokemons")
```

```
    plt.legend()
```

```
    plt.show()
```

```
else:
```

```
    print("enter valid input")
```

```
def Dragonbar():
```

```
    Dragon=pd.read_csv("D:\python projects\schoolproject\Bug.csv")
```

```
    Namei=Dragon["Name"]
```

```
    HPi=Dragon["HP"]
```

```
    Attacki=Dragon["Attack"]
```

```
    Defensei=Dragon["Defense"]
```

```
    plt.xlabel("Name of pokemon")
```

```
    ax=plt.axes()
```



```
print("press 1 for the data for name of pokemon Vs HP")
print("press 2 for the data for name of pokemon Vs Attack")
print("press 3 for the data for name of pokemon Vs Defense")
print("press 4 to merge all the data in one stackbar chart")
print("press 5 to merge all the data in one multibar chart")
```

```
pt=int(input("enter what type of graph u want to see:"))
```

```
if pt==1:
```

```
    dragonbar=ax.set_facecolor("navajowhite")
    plt.ylabel("HP of pokemons")
    plt.title("Name Vs HP")
    plt.bar(Namei,HPi)
    plt.xticks(rotation="vertical")
    plt.show()
```

```
elif pt==2:
```

```
    dragonbar=ax.set_facecolor("bisque")
    plt.ylabel("attaking strength of pokemons")
    plt.title("Name Vs Attack strength")
    plt.bar(Namei,Attacki,color="r")
    plt.xticks(rotation="vertical")
    plt.show()
```

```
elif pt==3:
```

```
    dragonbar=ax.set_facecolor("lightpink")
    plt.ylabel("defense strength of pokemons")
    plt.title("Name Vs defense strength")
    plt.bar(Namei,Defensei,color="m")
    plt.xticks(rotation="vertical")
```



```
plt.show()
```

```
elif pt==4:
```

```
    dragonbar=ax.set_facecolor("peachpuff")
```

```
    plt.ylabel("HP,Attack and defense strength")
```

```
    plt.title("Name Vs all abilities")
```

```
    plt.bar(Namei,HPi,width=0.2,label="Name Vs HP")
```

```
    plt.bar(Namei,Attacki,width=0.2,label="Name Vs Attack strength")
```

```
    plt.bar(Namei,Defensei,width=0.2,label="Name Vs defense strength")
```

```
    plt.xticks(rotation="vertical")
```

```
    plt.legend()
```

```
    plt.show()
```

```
elif pt==5:
```

```
    dragonbar=ax.set_facecolor("peachpuff")
```

```
    VD=np.arange(len(Namei))
```

```
    plt.bar(VD,HPi,width=0.33,label="Name Vs HP")
```

```
    plt.bar(VD+0.33,Attacki,width=0.33,label="Name Vs Attack strength")
```

```
    plt.bar(VD+0.66,Defensei,width=0.33,label="Name Vs defense strength")
```

```
    plt.xticks(VD,Namei,rotation="vertical")
```

```
    plt.title("Name Vs all abilities")
```

```
    plt.legend()
```

```
    plt.show()
```

```
else:
```

```
    print("enter valid input")
```

```
def Dragonhistogram():
```

```
    Dragon=pd.read_csv("D:\python  
projects\schoolproject\Bug.csv")
```

```
    Totali=Dragon["Total"]
```



```

Dragonhisto=ax.set_facecolor("mistyrose")

plt.hist(Totali,bins=[300,400,500,600],color="m")

plt.ylabel("Total Number of different DRAGON pokemons")

plt.title("Range wise total number of Dragon pokemons")

plt.xticks(rotation="vertical")

plt.show()

```

```

def Dragonline():

    Dragon=pd.read_csv("D:\python projects\schoolproject\Dragon.csv")

    Namei=Dragon["Name"]

    HPi=Dragon["HP"]

    Attacki=Dragon["Attack"]

    Defensei=Dragon["Defense"]

    plt.xlabel("Name of pokemon")

    plt.xticks(rotation="vertical")

    ax=plt.axes()


    print("press 1 to print the data for POKEMON v/s HP")

    print("press 2 to print the data for POKEMON v/s ATTACK")

    print("press 3 to print the data for POKEMON v/s DEFENCE")

    print("press 4 to print the data for POKEMON v/s ALL ABILITIES")


    pt=int(input("Enter your choice="))


    if pt==1:

        Dragonline=ax.set_facecolor("paleturquoise")

        plt.ylabel("HP of pokemons")

        plt.title("Name Vs HP")

```



```

plt.plot(Namei,HPi,color="k",marker='d',markersize=11,markeredgecolor='orchid',markerfacecolor="darkblue",linestyle="dashed",linewidth=2,label="NAME V/S HP")

plt.show()

elif pt==2:

    Dragonline=ax.set_facecolor("paleturquoise")

    plt.ylabel("attacking strength of pokemons")

    plt.title("Name Vs Attack strength")

plt.plot(Namei,Attacki,color="k",marker='d',markersize=11,markeredgecolor='orchid',markerfacecolor="darkblue",linestyle="dashed",linewidth=2,label="NAME V/S ATTACK")

plt.show()

elif pt==3:

    Darkline=ax.set_facecolor("paleturquoise")

    plt.ylabel("defense strength of pokemons")

    plt.title("Name Vs defense strength")

plt.plot(Namei,Defensei,color="k",marker='d',markersize=11,markeredgecolor='orchid',markerfacecolor="darkblue",linestyle="dashed",linewidth=2,label="NAME V/S DEFENSE")

plt.show()

elif pt==4:

    Dragonline=ax.set_facecolor("paleturquoise")

    plt.ylabel("HP,Attack and defense strength")

    plt.title("Name Vs abilities")

    plt.plot(Namei,HPi,color="b",label="NAME V/S HP")

    plt.plot(Namei,Attacki,color="k",label="NAME V/S ATTACK")

    plt.plot(Namei,Defensei,color="g",label="NAME V/S DEFENSE")

    plt.legend()

```



```

plt.show()

else:
    print("enter valid input")

def Dragonpie():
    Dragon=pd.read_csv("D:\python projects\schoolproject\Dragon.csv")
    Namei=Dragon["Name"]
    HPi=Dragon["HP"]
    Attacki=Dragon["Attack"]
    Defensei=Dragon["Defense"]

    print("press 1 for the data for name of pokemon Vs HP of pokemons")
    print("press 2 for the data for name of pokemon Vs Attack strength of pokemons")
    print("press 3 for the data for name of pokemon Vs Defense strength of pokemons")

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:
        plt.title("name vs HP")
        plt.pie(HPi,labels=Namei)
        plt.show()

    elif pt==2:
        plt.title("Name Vs Attack strength")
        plt.pie(Attacki,labels=Namei)
        plt.show()

    elif pt==3:
        plt.title("Name Vs defense strength")
        plt.pie(Defensei,labels=Namei)

```



```

plt.show()

else:
    print("enter vaid input")

def Dragonscatter():
    Dragon=pd.read_csv("D:\python projects\schooolproject\Dragon.csv")
    Namei=Dragon["Name"]
    HPi=Dragon["HP"]
    Attacki=Dragon["Attack"]
    Defensei=Dragon["Defense"]
    plt.xlabel("Name of pokemons")
    plt.xticks(rotation="vertical")

    print("press 1 for the data for name of pokemon Vs HP")
    print("press 2 for the data for name of pokemon Vs Attack")
    print("press 3 for the data for name of pokemon Vs Defense")
    print("press 4 to merge all the data in one scatter chart")

    ax=plt.axes()
    axs=plt.gca()

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:
        Dragonscatter=ax.set_facecolor("paleturquoise")
        plt.ylabel("HP of pokemons")
        ax.scatter(Namei,HPi,color="r")
        plt.title("Name vs HP")
        plt.show()

```




```

elif pt==2:
    Dragonscatter=ax.set_facecolor("paleturquoise")
    plt.ylabel("Attack strength of pokemons")
    ax.scatter(Namei,Attacki,color="b")
    plt.title("Name vs Attack strength")
    plt.show()

elif pt==3:
    Dragonscatter=ax.set_facecolor("paleturquoise")
    plt.ylabel("Defense strength of pokemons")
    ax.scatter(Namei,Defensei,color="darkgreen")
    plt.title("Name Vs defense strength")
    plt.show()

elif pt==4:
    Dragonscatter=ax.set_facecolor("paleturquoise")
    plt.ylabel("All abilities of pokemons")
    ax.scatter(Namei,HPi,color="r",label="Name Vs HP")
    ax.scatter(Namei,Attacki,color="b",label="Name Vs Attack")
    ax.scatter(Namei,Defensei,color="darkgreen",label="Name Vs Defense")
    plt.title("complete analysis of pokemons")
    plt.legend()
    plt.show()

else:
    print("enter valid input")

```

```

def Electricbar():
    Electric=pd.read_csv("D:\python
projects\schoolproject\Bug.csv")
    Namei=Electric["Name"]

```



```

HPi=Electric["HP"]
Attacki=Electric["Attack"]
Defensei=Electric["Defense"]
plt.xlabel("Name of pokemon")
ax=plt.axes()

print("press 1 for the data for name of pokemon Vs HP")
print("press 2 for the data for name of pokemon Vs Attack")
print("press 3 for the data for name of pokemon Vs Defense")
print("press 4 to merge all the data in one stackbar chart")
print("press 5 to merge all the data in one multibar chart")

pt=int(input("enter what type of graph u want to see:"))

if pt==1:
    electricbar=ax.set_facecolor("navajowhite")
    plt.ylabel("HP of pokemons")
    plt.title("Name Vs HP")
    plt.bar(Namei,HPi)
    plt.xticks(rotation="vertical")
    plt.show()

elif pt==2:
    electricbar=ax.set_facecolor("bisque")
    plt.ylabel("attaking strength of pokemons")
    plt.title("Name Vs Attack strength")
    plt.bar(Namei,Attacki,color="r")
    plt.xticks(rotation="vertical")
    plt.show()

elif pt==3:

```



```
electricbar=ax.set_facecolor("lightpink")  
plt.ylabel("defense strength of pokemons")  
plt.title("Name Vs defense strength")  
plt.bar(Namei,Defensei,color="m")  
plt.xticks(rotation="vertical")  
plt.show()
```

```
elif pt==4:
```

```
electricbar=ax.set_facecolor("peachpuff")  
plt.ylabel("HP,Attack and defense strength")  
plt.title("Name Vs abilities")  
plt.bar(Namei,HPi,width=0.2,label="Name Vs HP")  
plt.bar(Namei,Attacki,width=0.2,label="Name Vs Attack strength")  
plt.bar(Namei,Defensei,width=0.2,label="Name Vs defense strength")  
plt.xticks(rotation="vertical")  
plt.legend()  
plt.show()
```

```
elif pt==5:
```

```
electricbar=ax.set_facecolor("peachpuff")  
VD=np.arange(len(Namei))  
plt.bar(VD,HPi,width=0.33,label="Name Vs HP")  
plt.bar(VD+0.33,Attacki,width=0.33,label="Name Vs Attack strength")  
plt.bar(VD+0.66,Defensei,width=0.33,label="Name Vs defense strength")  
plt.xticks(VD,Namei,rotation="vertical")  
plt.legend()  
plt.show()
```

```
else:
```

```
print("enter valid input")
```



```

def Electrichistogram():
    Electric=pd.read_csv("D:\python projects\schoolproject\Bug.csv")
    Totali=Electric["Total"]
    Electrichisto=ax.set_facecolor("cornsilk")
    plt.hist(Totali,bins=[300,400,500,600],color="g")
    plt.ylabel("Total Number of different ELECTRIC pokemons")
    plt.title("Range wise total number of electric pokemons")
    plt.xticks(rotation="vertical")
    plt.show()

def Electricline():
    electric=pd.read_csv("D:\python projects\schoolproject\Electric.csv")
    Namei=electric["Name"]
    HPi=electric["HP"]
    Attacki=electric["Attack"]
    Defensei=electric["Defense"]
    plt.xlabel("Name of pokemon")
    plt.xticks(rotation="vertical")
    ax=plt.axes()

    print("press 1 for the data for name of pokemon Vs HP")
    print("press 2 for the data for name of pokemon Vs Attack")
    print("press 3 for the data for name of pokemon Vs Defense")
    print("press 4 to merge all the data in one line chart")

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:
        electricline=ax.set_facecolor("khaki")

```



```

plt.ylabel("HP of pokemons")

plt.title("Name Vs HP")

plt.plot(Namei,HPi,"k",marker="s",markersize=8,markerfacecolor="peru",markeredgecolor=
"black",linestyle="dashdot",linewidth=2)

plt.show()

elif pt==2:

    electricline=ax.set_facecolor("khaki")

    plt.ylabel("attaking strength of pokemons")

    plt.title("Name Vs Attack strength")

plt.plot(Namei,Attacki,"k",marker="s",markersize=8,markerfacecolor="peru",markeredgecolor=
"black",linestyle="dashdot",linewidth=2)

plt.show()

elif pt==3:

    electricline=ax.set_facecolor("khaki")

    plt.ylabel("defense strength of pokemons")

    plt.title("Name Vs defense strength")

plt.plot(Namei,Defensei,"k",marker="s",markersize=8,markerfacecolor="peru",markeredgecolor=
"black",linestyle="dashdot",linewidth=2)

plt.show()

elif pt==4:

    electricline=ax.set_facecolor("khaki")

    plt.ylabel("HP,Attack and defense strength")

    plt.title("Name Vs abilities")

    plt.plot(Namei,HPi,label="Name Vs HP")

    plt.plot(Namei,Attacki,label="Name Vs Attack strength")

    plt.plot(Namei,Defensei,label="Name Vs defense strength")

```



```

plt.legend()

plt.show()

else:
    print("enter valid input")

def Electricpie():
    Electric=pd.read_csv("D:\python projects\schoolproject\Electric.csv")
    Namei=Electric["Name"]
    HPi=Electric["HP"]
    Attacki=Electric["Attack"]
    Defensei=Electric["Defense"]

    print("press 1 for the data for name of pokemon Vs HP of pokemons")
    print("press 2 for the data for name of pokemon Vs Attack strength of pokemons")
    print("press 3 for the data for name of pokemon Vs Defense strength of pokemons")

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:
        plt.title("name vs HP")
        plt.pie(HPi,labels=Namei)
        plt.show()

    elif pt==2:
        plt.title("Name Vs Attack strength")
        plt.pie(Attacki,labels=Namei)
        plt.show()

    elif pt==3:
        plt.title("Name Vs defense strength")

```



```

plt.pie(Defensei,labels=Namei)

plt.show()

else:

    print("enter vaid input")

def Electricscatter():

    Electric=pd.read_csv("D:\python projects\schoolproject\Electric.csv")

    Namei=Electric["Name"]

    HPi=Electric["HP"]

    Attacki=Electric["Attack"]

    Defensei=Electric["Defense"]

    plt.xlabel("Name of pokemons")

    plt.xticks(rotation="vertical")


    print("press 1 for the data for name of pokemon Vs HP")

    print("press 2 for the data for name of pokemon Vs Attack")

    print("press 3 for the data for name of pokemon Vs Defense")

    print("press 4 to merge all the data in one scatter chart")


    ax=plt.axes()

    axs=plt.gca()


    pt=int(input("enter what type of graph u want to see:"))


    if pt==1:

        electricscatter=ax.set_facecolor("papayawhip")

        axs.scatter(Namei,HPi,color="chocolate")

        plt.title("Name vs HP")

        plt.ylabel("HP of pokemons")

        plt.show()

```



```
elif pt==2:
    electricscatter=ax.set_facecolor("papayawhip")
    axs.scatter(Namei,Attacki,color="limegreen")
    plt.title("Name vs Attack strength")
    plt.ylabel("attack strength of pokemons")
    plt.show()

elif pt==3:
    electricscatter=ax.set_facecolor("papayawhip")
    axs.scatter(Namei,Defensei,color="cornflowerblue")
    plt.title("Name Vs defense strength")
    plt.ylabel("defense strength of pokemons")
    plt.show()

elif pt==4:
    electricscatter=ax.set_facecolor("papayawhip")
    axs.scatter(Namei,HPi,color="chocolate",label="Name Vs HP")
    axs.scatter(Namei,Attacki,color="limegreen",label="Name Vs Attack")
    axs.scatter(Namei,Defensei,color="cornflowerblue",label="Name Vs Defense")
    plt.title("complete analysis of pokemons")
    plt.ylabel("all abilities of pokemons")
    plt.legend()
    plt.show()

else:
    print("enter valid input")
```

```
def Fairybar():
```




```

Fairy=pd.read_csv("D:\python projects\schoolproject\Bug.csv")
Namei=Fairy["Name"]
HPi=Fairy["HP"]
Attacki=Fairy["Attack"]
Defensei=Fairy["Defense"]
plt.xlabel("Name of pokemon")
ax=plt.axes()

print("press 1 for the data for name of pokemon Vs HP")
print("press 2 for the data for name of pokemon Vs Attack")
print("press 3 for the data for name of pokemon Vs Defense")
print("press 4 to merge all the data in one stackbar chart")
print("press 5 to merge all the data in one multibar chart")

pt=int(input("enter what type of graph u want to see:"))

if pt==1:
    fairybar=ax.set_facecolor("navajowhite")
    plt.ylabel("HP of pokemons")
    plt.title("Name Vs HP")
    plt.barh(Namei,HPi)
    plt.xticks(rotation="vertical")
    plt.show()

elif pt==2:
    fairybar=ax.set_facecolor("bisque")
    plt.ylabel("attaking strength of pokemons")
    plt.title("Name Vs Attack strength")
    plt.barh(Namei,Attacki,color="r")
    plt.xticks(rotation="vertical")
    plt.show()

```



```

elif pt==3:
    fairybar=ax.set_facecolor("lightpink")
    plt.ylabel("defense strength of pokemons")
    plt.title("Name Vs defense strength")
    plt.barh(Namei,Defensei,color="m")
    plt.xticks(rotation="vertical")
    plt.show()

elif pt==4:
    fairybar=ax.set_facecolor("peachpuff")
    plt.ylabel("HP,Attack and defense strength")
    plt.title("Name Vs abilities")
    plt.bar(Namei,HPi,width=0.2,label="Name Vs HP")
    plt.bar(Namei,Attacki,width=0.2,label="Name Vs Attack strength")
    plt.bar(Namei,Defensei,width=0.2,label="Name Vs defense strength")
    plt.xticks(rotation="vertical")
    plt.legend()
    plt.show()

elif pt==5:
    fairybar=ax.set_facecolor("peachpuff")
    VD=np.arange(len(Namei))
    plt.bar(VD,HPi,width=0.33,label="Name Vs HP")
    plt.bar(VD+0.33,Attacki,width=0.33,label="Name Vs Attack strength")
    plt.bar(VD+0.66,Defensei,width=0.33,label="Name Vs defense strength")
    plt.xticks(VD,Namei,rotation="vertical")
    plt.legend()
    plt.show()

```

```

else:

```



```
print("enter valid input")
```

```
def Fairyhistogram():
```

```
    Fairy=pd.read_csv("D:\python projects\schoolproject\Fairy.csv")
```

```
    Totali=Fairy["Total"]
```

```
    Fairyhisto=ax.set_facecolor("antiquewhite")
```

```
    plt.hist(Totali,bins=[300,400,500,600],color="c")
```

```
    plt.ylabel("Total Number of different FAIRY pokemons")
```

```
    plt.title("Range wise total number of Fairy pokemons")
```

```
    plt.xticks(rotation="vertical")
```

```
    plt.show()
```

```
def Fairyline():
```

```
    Fairy=pd.read_csv("D:\python projects\schoolproject\Fairy.csv")
```

```
    Namei=Fairy["Name"]
```

```
    HPi=Fairy["HP"]
```

```
    Attacki=Fairy["Attack"]
```

```
    Defensei=Fairy["Defense"]
```

```
    plt.xlabel("Name of pokemon")
```

```
    plt.xticks(rotation="vertical")
```

```
    ax=plt.axes()
```

```
print("press 1 for the data for name of pokemon Vs HP")
```

```
print("press 2 for the data for name of pokemon Vs Attack")
```

```
print("press 3 for the data for name of pokemon Vs Defense")
```

```
print("press 4 to merge all the data in one line chart")
```

```
pt=int(input("enter what type of graph u want to see:"))
```



```

if pt==1:

    Fairyline=ax.set_facecolor("peachpuff")

    plt.ylabel("HP of pokemons")

    plt.title("Name Vs HP")

plt.plot(Namei,HPi,"r",marker="D",markersize=9,markerfacecolor="maroon",markeredgecolor="royalblue",linestyle="solid",linewidth=2)

    plt.show()

elif pt==2:

    Fairyline=ax.set_facecolor("peachpuff")

    plt.ylabel("attacking strength of pokemons")

    plt.title("Name Vs Attack strength")

plt.plot(Namei,Attacki,"r",marker="D",markersize=9,markerfacecolor="maroon",markeredgecolor="royalblue",linestyle="solid",linewidth=2)

    plt.show()

elif pt==3:

    Fairyline=ax.set_facecolor("peachpuff")

    plt.ylabel("defense strength of pokemons")

    plt.title("Name Vs defense strength")

plt.plot(Namei,Defensei,"r",marker="D",markersize=9,markerfacecolor="maroon",markeredgecolor="royalblue",linestyle="solid",linewidth=2)

    plt.show()

elif pt==4:

    Fairyline=ax.set_facecolor("cornsilk")

    plt.ylabel("HP,Attack and defense strength")

    plt.title("Name Vs abilities")

    plt.plot(Namei,HPi,label="Name Vs HP")

    plt.plot(Namei,Attacki,label="Name Vs Attack strength")

```



```

plt.plot(Namei,Defensei,label="Name Vs defense strength")

plt.legend()

plt.show()

else:

    print("enter valid input")

def Fairypie():

    Fairy=pd.read_csv("D:\python projects\schoolproject\Fairy.csv")

    Namei=Fairy["Name"]

    HPi=Fairy["HP"]

    Attacki=Fairy["Attack"]

    Defensei=Fairy["Defense"]

    print("press 1 for the data for name of pokemon Vs HP of pokemons")

    print("press 2 for the data for name of pokemon Vs Attack strength of pokemons")

    print("press 3 for the data for name of pokemon Vs Defense strength of pokemons")

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:

        plt.title("name vs HP")

        plt.pie(HPi,labels=Namei)

        #autopct attribute can be used to show the percentage of the data.

        plt.show()

    elif pt==2:

        plt.title("Name Vs Attack strength")

        plt.pie(Attacki,labels=Namei)

        plt.show()

    elif pt==3:

        plt.title("Name Vs defense strength")

```



```

plt.pie(Defensei,labels=Namei)

plt.show()

else:

    print("enter vaid input")

def Fairyscatter():

    Fairy=pd.read_csv("D:\python projects\schoolproject\Fairy.csv")

    Namei=Fairy["Name"]

    HPi=Fairy["HP"]

    Attacki=Fairy["Attack"]

    Defensei=Fairy["Defense"]

    plt.xlabel("Name of pokemons")

    plt.xticks(rotation="vertical")


    print("press 1 for the data for name of pokemon Vs HP")

    print("press 2 for the data for name of pokemon Vs Attack")

    print("press 3 for the data for name of pokemon Vs Defense")

    print("press 4 to merge all the data in one scatter chart")


    ax=plt.axes()

    axs=plt.gca()


    pt=int(input("enter what type of graph u want to see:"))


    if pt==1:

        Fairyscatter=ax.set_facecolor("bisque")

        axs.scatter(Namei,HPi,color="r")

        plt.ylabel("Hp of pokemons")

        plt.title("Name vs HP")

        plt.show()

```



```

elif pt==2:
    Fairyscatter=ax.set_facecolor("bisque")
    axs.scatter(Namei,Attacki,color="m")
    plt.ylabel("attack strength of pokemons")
    plt.title("Name vs Attack strength")
    plt.show()

elif pt==3:
    Fairyscatter=ax.set_facecolor("bisque")
    axs.scatter(Namei,Defensei,color="teal")
    plt.ylabel("defense strength of pokemons")
    plt.title("Name Vs defense strength")
    plt.show()

elif pt==4:
    Fairyscatter=ax.set_facecolor("bisque")
    axs.scatter(Namei,HPi,color="r",label="Name Vs HP")
    axs.scatter(Namei,Attacki,color="m",label="Name Vs Attack")
    axs.scatter(Namei,Defensei,color="teal",label="Name Vs Defense")
    plt.ylabel("all abilities")
    plt.title("complete analysis of pokemons")
    plt.legend()
    plt.show()

else:
    print("enter valid input")

```

```

def Fightingbar():
    Fighting=pd.read_csv("D:\python
projects\schoolproject\Bug.csv")

    Namei=Fighting["Name"]
    HPi=Fighting["HP"]

```



```

Attacki=Fighting["Attack"]
Defensei=Fighting["Defense"]
plt.xlabel("Name of pokemon")
ax=plt.axes()

print("press 1 for the data for name of pokemon Vs HP")
print("press 2 for the data for name of pokemon Vs Attack")
print("press 3 for the data for name of pokemon Vs Defense")
print("press 4 to merge all the data in one stackbar chart")
print("press 5 to merge all the data in one multibar chart")

pt=int(input("enter what type of graph u want to see:"))

if pt==1:
    fightingbar=ax.set_facecolor("navajowhite")
    plt.ylabel("HP of pokemons")
    plt.title("Name Vs HP")
    plt.bar(Namei,HPi)
    plt.xticks(rotation="vertical")
    plt.show()

elif pt==2:
    fightingbar=ax.set_facecolor("bisque")
    plt.ylabel("attaking strength of pokemons")
    plt.title("Name Vs Attack strength")
    plt.bar(Namei,Attacki,color="r")
    plt.xticks(rotation="vertical")
    plt.show()

elif pt==3:
    fightingbar=ax.set_facecolor("lightpink")

```




```
plt.ylabel("defense strength of pokemons")
plt.title("Name Vs defense strength")
plt.bar(Namei,Defensei,color="m")
plt.xticks(rotation="vertical")
plt.show()
```

```
elif pt==4:
```

```
    fightingbar=ax.set_facecolor("peachpuff")
    plt.ylabel("HP,Attack and defense strength")
    plt.title("Name Vs abilities")
    plt.bar(Namei,HPi,width=0.2,label="Name Vs HP")
    plt.bar(Namei,Attacki,width=0.2,label="Name Vs Attack strength")
    plt.bar(Namei,Defensei,width=0.2,label="Name Vs defense strength")
    plt.xticks(rotation="vertical")
    plt.legend()
    plt.show()
```

```
elif pt==5:
```

```
    fightingbar=ax.set_facecolor("peachpuff")
    VD=np.arange(len(Namei))
    plt.bar(VD,HPi,width=0.33,label="Name Vs HP")
    plt.bar(VD+0.33,Attacki,width=0.33,label="Name Vs Attack strength")
    plt.bar(VD+0.66,Defensei,width=0.33,label="Name Vs defense strength")
    plt.xticks(VD,Namei,rotation="vertical")
    plt.legend()
    plt.show()
```

```
else:
```

```
    print("enter valid input")
```

```
def Fightinghistogram():
```



```

Fighting=pd.read_csv("D:\python projects\schoolproject\Bug.csv")
Totali=Fighting["Total"]
Fightinghisto=ax.set_facecolor("lightblue")
plt.hist(Totali,bins=[300,400,500,600],color="r")
plt.ylabel("Total Number of different Fighting pokemons")
plt.title("Range wise total number of fighting pokemons")
plt.xticks(rotation="vertical")
plt.show()

```

```
def Fightingline():
```

```

    Fighting=pd.read_csv("D:\python projects\schoolproject\Fighting.csv")
    Namei=Fighting["Name"]
    HPi=Fighting["HP"]
    Attacki=Fighting["Attack"]
    Defensei=Fighting["Defense"]
    plt.xlabel("Name of pokemon")
    plt.xticks(rotation="vertical")
    ax=plt.axes()

```

```

print("press 1 for the data for name of pokemon Vs HP")
print("press 2 for the data for name of pokemon Vs Attack")
print("press 3 for the data for name of pokemon Vs Defense")
print("press 4 to merge all the data in one line chart")

```

```
pt=int(input("enter what type of graph u want to see:"))
```

```
if pt==1:
```

```

    Fightingline=ax.set_facecolor("blanchedalmond")
    plt.ylabel("HP of pokemons")
    plt.title("Name Vs HP")

```



```

plt.plot(Namei,HPi,"m",marker="h",markersize=11,markerfacecolor="blue",markeredgecolor="darkgreen",linestyle="solid",linewidth=3)

plt.show()

elif pt==2:

    Fightingline=ax.set_facecolor("blanchedalmond")

    plt.ylabel("attacking strength of pokemons")

    plt.title("Name Vs Attack strength")

plt.plot(Namei,Attacki,"m",marker="h",markersize=11,markerfacecolor="blue",markeredgecolor="darkgreen",linestyle="solid",linewidth=3)

plt.show()

elif pt==3:

    Fightingline=ax.set_facecolor("blanchedalmond")

    plt.ylabel("defense strength of pokemons")

    plt.title("Name Vs defense strength")

plt.plot(Namei,Defensei,"m",marker="h",markersize=11,markerfacecolor="blue",markeredgecolor="darkgreen",linestyle="solid",linewidth=3)

plt.show()

elif pt==4:

    Fightingline=ax.set_facecolor("lemonchiffon")

    plt.ylabel("HP,Attack and defense strength")

    plt.title("Name Vs abilities")

    plt.plot(Namei,HPi,label="Name Vs HP")

    plt.plot(Namei,Attacki,label="Name Vs Attack strength")

    plt.plot(Namei,Defensei,label="Name Vs defense strength")

    plt.legend()

    plt.show()

```



```

else:
    print("enter valid input")

def Fightingpie():
    Fighting=pd.read_csv("D:\python projects\schoolproject\Fighting.csv")
    Namei=Fighting["Name"]
    HPi=Fighting["HP"]
    Attacki=Fighting["Attack"]
    Defensei=Fighting["Defense"]
    ax=plt.axes()

    print("press 1 for the data for name of pokemon Vs HP of pokemons")
    print("press 2 for the data for name of pokemon Vs Attack strength of pokemons")
    print("press 3 for the data for name of pokemon Vs Defense strength of pokemons")

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:
        plt.title("name vs HP")
        plt.pie(HPi,labels=Namei)
        #autopct attribute can be used to show the percentage of the data.
        plt.show()
    elif pt==2:
        plt.title("Name Vs Attack strength")
        plt.pie(Attacki,labels=Namei)
        plt.show()
    elif pt==3:
        plt.title("Name Vs defense strength")
        plt.pie(Defensei,labels=Namei)
        plt.show()
    else:

```



```

    print("enter void input")

def Fightingscatter():
    Fighting=pd.read_csv("D:\python projects\schoolproject\Fighting.csv")
    Namei=Fighting["Name"]
    HPi=Fighting["HP"]
    Attacki=Fighting["Attack"]
    Defensei=Fighting["Defense"]
    plt.xlabel("Name of pokemons")
    plt.xticks(rotation="vertical")

    print("press 1 for the data for name of pokemon Vs HP")
    print("press 2 for the data for name of pokemon Vs Attack")
    print("press 3 for the data for name of pokemon Vs Defense")
    print("press 4 to merge all the data in one scatter chart")

    ax=plt.axes()
    axs=plt.gca()

    pt=int(input("enter what type of graph u want to see:"))

    if pt==1:
        Fightingscatter=ax.set_facecolor("mistyrose")
        axs.scatter(Namei,HPi,color="blue")
        plt.ylabel("HP of Pokemons")
        plt.title("Name vs HP")
        plt.show()

    elif pt==2:
        Fightingscatter=ax.set_facecolor("mistyrose")
        axs.scatter(Namei,Attacki,color="crimson")

```



```

plt.title("Name vs Attack strength")

plt.ylabel("Attack strength of Pokemons")

plt.show()

elif pt==3:

    Fightingscatter=ax.set_facecolor("mistyrose")

    axs.scatter(Namei,Defensei,color="dodgerblue")

    plt.ylabel("Defense strength of Pokemons")

    plt.title("Name Vs defense strength")

    plt.show()

elif pt==4:

    Fightingscatter=ax.set_facecolor("mistyrose")

    axs.scatter(Namei,HPi,color="blue",label="Name Vs HP")

    axs.scatter(Namei,Attacki,color="crimson",label="Name Vs Attack")

    axs.scatter(Namei,Defensei,color="dodgerblue",label="Name Vs Defense")

    plt.ylabel("all abilities of pokemons")

    plt.title("complete analysis of pokemons")

    plt.legend()

    plt.show()

else:

    print("enter valid input")

```

```

#####          graphs          #####

```

```

#####          functions          #####

```

```

#ascending- descending function

```

```

f=pd.read_csv("D:\python projects\schoolproject\Pokemon.csv")

```

```

def ascdsc():

```



```

print("type A for ascending order or type D for descending order")

inp1=input("enter your input=")

print("a is for Name")
print("b is for HP")
print("c is for attack")
print("d for defense")
print("e is for speed")

inp2=input("enter the column according to which you want to see records=")

if inp1=='A':
    if inp2=='a':
        f.sort_values(['Name'],inplace=True)
        print(f)
    elif inp2=="b":
        f.sort_values(['HP'],inplace=True)
        print(f)
    elif inp2=="c":
        f.sort_values(['Attack'],inplace=True)
        print(f)
    elif inp2=="d":
        f.sort_values(['Defense'],inplace=True)
        print(f)
    elif inp2=="e":
        f.sort_values(['Speed'],inplace=True)
        print(f)
elif inp1=="D":
    if inp2=='a':

f.sort_values(['Name'],ascending=False,inplace=True)
        print(f)
    elif inp2=='b':

```



```

        f.sort_values(['HP'],ascending=False,inplace=True)

        print(f)

    elif inp2=='c':

        f.sort_values(['Attack'],ascending=False,inplace=True)

        print(f)

    elif inp2=='d':

        f.sort_values(['Defense'],ascending=False,inplace=True)

        print(f)

    elif inp2=='e':

        f.sort_values(['Speed'],ascending=False,inplace=True)

        print(f)

    else:

        print("enter valid input")

        print("you will get another chance")


#for going to next type
nxttype="okay we will take you to next type"


#proceed function
def proceed():

    proceed1=input("enter y to proceed(y/n)")

    while proceed1 != "y":

        print("press y to go ahead")

        proceed1=input("enter y to proceed(y/n)")

        if proceed1=="y":

            break


#Bug function
def Bug():

    print ("Reading Data from File in Different way")

    print ("A: Read csv file")

```




```

print ("B: Reading file without index")
print(" ")
print ("Manipulation data in the records of csv File")
print ("C: Sort the data as per ascending or descending order")
print ("D: Read Top and Bottom Records file as per requirement")
print ("E: Make the copy of csv file")
print ("F: lead the Specific column")
print(" ")
print ("Data Visualization")
print ("1: Line Chart")
print ("2: Bar Plot")
print ("3: Pie chart")
print ("4: Scatter chart")
print("5: Histogram")
print(" ")
bugfile=pd.read_csv("D:\python projects\schoolproject\Bug.csv")
f=bugfile

def Bugif():
    inp=input("please enter your from above given options=")
    if inp=="A":
        print(bugfile)
    elif inp=="B":
        print(pd.read_csv("D:\python projects\schoolproject\Bug.csv",index_col=0))
    elif inp=="C":
        ascdsc()
    elif inp=="D":
        Dinput=input("enter T for seeing top racords/B for
bottom records=")
        if Dinput=="T":
            print("you have selected to see top records")

```



```

        DTinput=int(input("enter number of records you want to see="))

        print(bugfile.head(DTinput))

    elif Dinput==B:

        print("you have selected to see bottom records")

        DBinput=int(input("enter number of records you want to see="))

        print(bugfile.tail(DBinput))

    else:

        print("enter valid input")

elif inp=="E":

    print("we made copy of csv file and it is printed")

    bugnew=bugfile.to_csv("D:\python projects\schoolproject\Bugnew.csv")

    print(pd.read_csv("D:\python projects\schoolproject\Bugnew.csv"))

elif inp=="F":

    df=pd.read_csv("D:\python
projects\schoolproject\Bug.csv",usecols=['Name','Total'],index_col=0)

    print(df)

elif inp=="1":

    Bugline()

elif inp=="2":

    Bugbar()

elif inp=="3":

    Bugpie()

elif inp=="4":

    Bugscatter()

elif inp=="5":

    Bughistogram()

else:

    print("please enter input from above given
options")

# call for taking input for first time

Bugif()

inp2=input("want to do it again(yes/no)=")

```



```

while inp2=="yes":
    Bugif()
    inp2=input("want to do it again(yes/no)=")
    if inp2 == "no":
        break
    else:
        print("please enter proper input")

#-----
#Dark function
def Dark():
    print ("Reading Data from File in Different way")
    print ("A: Read csv file")
    print ("B: Reading file without index")
    print(" ")
    print ("Apply Data Manipulation in the records of csv File")
    print ("C: Sorting the data as per your choice")
    print ("D: Read Top and Bottom Records file as per requirement")
    print ("E: Make the copy of csv file")
    print ("F: lead the Specific column")
    print(" ")
    print ("Data Visualization")
    print ("1: Line Chart")
    print ("2: Bar Plot")
    print ("3: Pie chart")
    print ("4: Scatter chart")
    print("5: Histogram")
    print("dark function")

    darkfile=pd.read_csv("D:\python
projects\schoolproject\Dark.csv")

    f=darkfile

```



```

def Darkif():
    inp=input("please enter your  from above given options=")
    if inp=="A":
        print(darkfile)
    elif inp=="B":
        print(pd.read_csv("D:\python
projects\schoolproject\Dark.csv",index_col=0))
    elif inp=="C":
        ascdec()
    elif inp=="D":
        Dinput=input("enter T for seeing top racords/B for bottom records=")
        if Dinput=="T":
            print("you have selected to see top records")
            DTinput=int(input("enter number of records you want to see="))
            print(darkfile.head(DTinput))
        elif Dinput=="B":
            print("you have selected to see bottom records")
            DBinput=int(input("enter number of records you want to see="))
            print(darkfile.tail(DBinput))
        else:
            print("enter valid input")
    elif inp=="E":
        print("we made copy of csv file and it is printed")
        darknew=darkfile.to_csv("D:\python projects\schoolproject\Darknew.csv")
        print(pd.read_csv("D:\python projects\schoolproject\Darknew.csv"))
    elif inp=="F":

```



```

df=pd.read_csv("D:\python
projects\schoolproject\Dark.csv",usecols=['Name','Total'],index_col=0)

print(df)

elif inp=="1":

    Darkline()

elif inp=="2":

    Darkbar()

elif inp=="3":

    Darkpie()

elif inp=="4":

    Darkscatter()

elif inp=="5":

    Darkhistogram()

else:

    print("please enter input from above given options")

# call for taking input for first time

Darkif()

inp2=input("want to do it again(yes/no)=")

while inp2=="yes":

    Darkif()

    inp2=input("want to do it again(yes/no)=")

    if inp2 == "no":

        break

    else:

        print("please enter proper input")

#-----

#dragon function

def Dragon():

    print ("Reading Data from File in Different way")

    print ("A: Read csv file")

```



```

print ("B: Reading file without index")
print(" ")
print ("Apply Data Manipulation in the records of csv File")
print ("C: Sorting the data as per your choice")
print ("D: Read Top and Bottom Records file as per requirement")
print ("E: Make the copy of csv file")
print ("F: lead the Specific column")
print(" ")
print ("Data Visualization")
print ("1: Line Chart")
print ("2: Bar Plot")
print ("3: Pie chart")
print ("4: Scatter chart")
print("5: Histogram")
dragonfile=pd.read_csv("D:\python projects\schoolproject\Dragon.csv")

def Dragonif():
    f=dragonfile
    inp=input("please enter your  from above given options=")
    if inp=="A":
        print(dragonfile)
    elif inp=="B":
        print(pd.read_csv("D:\python
projects\schoolproject\Dragon.csv",index_col=0))
    elif inp=="C":
        ascdsc()
    elif inp=="D":
        Dinput=input("enter T for seeing top racords/B for
bottom records=")
        if Dinput=="T":
            print("you have selected to see top records")

```



```

        DTinput=int(input("enter number of records you want to see="))
        print(dragonfile.head(DTinput))
    elif Dinput=="B":
        print("you have selected to see bottom records")
        DBinput=int(input("enter number of records you want to see="))
        print(dragonfile.tail(DBinput))
    else:
        print("enter valid input")
elif inp=="E":
    print("we made copy of csv file and it is printed")
    dragonnew=dragonfile.to_csv("D:\python
projects\schoolproject\Dragonnew.csv")
    print(pd.read_csv("D:\python projects\schoolproject\Dragonnew.csv"))
elif inp=="F":
    df=pd.read_csv("D:\python
projects\schoolproject\Dragon.csv",usecols=['Name','Total'],index_col=0)
    print(df)
elif inp=="1":
    Dragonline()
elif inp=="2":
    Dragonbar()
elif inp=="3":
    Dragonpie()
elif inp=="4":
    Dragonscatter()
elif inp=="5":
    Dragonhistogram()
else:
    print("please enter input from above given
options")
# call for taking input for first time
    Dragonif()

```



```

inp2=input("want to do it again(yes/no)=")
while inp2=="yes":
    Dragonif()
    inp2=input("want to do it again(yes/no)=")
    if inp2 == "no":
        break
    else:
        print("please enter proper input")

#-----
#electric function
def Electric():
    print ("Reading Data from File in Different way")
    print ("A: Read csv file")
    print ("B: Reading file without index")
    print(" ")
    print ("Apply Data Manipulation in the records of csv File")
    print ("C: Sorting the data as per your choice")
    print ("D: Read Top and Bottom Records file as per requirement")
    print ("E: Make the copy of csv file")
    print ("F: lead the Specific column")
    print(" ")
    print ("Data Visualization")
    print ("1: Line Chart")
    print ("2: Bar Plot")
    print ("3: Pie chart")
    print ("4: Scatter chart")
    print("5: Histogram")

    electricfile=pd.read_csv("D:\python
projects\schoolproject\Electric.csv")

    f=electricfile

```




```

def Electricif():
    inp=input("please enter your  from above given options=")
    if inp=="A":
        print(electricfile)
    elif inp=="B":
        print(pd.read_csv("D:\python
projects\schoolproject\Electric.csv",index_col=0))
    elif inp=="C":
        ascdsc()
    elif inp=="D":
        Dinput=input("enter T for seeing top racords/B for bottom records=")
        if Dinput=="T":
            print("you have selected to see top records")
            DTinput=int(input("enter number of records you want to see="))
            print(electricfile.head(DTinput))
        elif Dinput=="B":
            print("you have selected to see bottom records")
            DBinput=int(input("enter number of records you want to see="))
            print(electricfile.tail(DBinput))
        else:
            print("enter valid input")
    elif inp=="E":
        print("we made copy of csv file and it is printed")
        electricnew=electricfile.to_csv("D:\python
projects\schoolproject\Electricnew.csv")
        print(pd.read_csv("D:\python projects\schoolproject\Electricnew.csv"))
    elif inp=="F":

```



```

df=pd.read_csv("D:\python
projects\schoolproject\Electric.csv",usecols=['Name','Total'],index_col=0)

print(df)

elif inp=="1":

    Electricline()

elif inp=="2":

    Electricbar()

elif inp=="3":

    Electricpie()

elif inp=="4":

    Electricscatter()

elif inp=="5":

    Electrichistogram()

else:

    print("please enter input from above given options")

# call for taking input for first time

Electricif()

inp2=input("want to do it again(yes/no)=")

while inp2=="yes":

    Electricif()

    inp2=input("want to do it again(yes/no)=")

    if inp2 == "no":

        break

    else:

        print("please enter proper input")

#-----

#fairy function

def Fairy():

    print ("Reading Data from File in Different way")

    print ("A: Read csv file")

```



```

print ("B: Reading file without index")

print("    ")

print ("Apply Data Manipulation in the records of csv File")

print ("C: Sorting the data as per your choice")

print ("D: Read Top and Bottom Records file as per requirement")

print ("E: Make the copy of csv file")

print ("F: lead the Specific column")

print("    ")

print ("Data Visualization")

print ("1: Line Chart")

print ("2: Bar Plot")

print ("3: Pie chart")

print ("4: Scatter chart")

print("5: Histogram")

fairyfile=pd.read_csv("D:\python projects\schoolproject\Fairy.csv")

f=fairyfile


def Fairyif():

    inp=input("please enter your  from above given options=")

    if inp=="A":

        print(fairyfile)

    elif inp=="B":

        print(pd.read_csv("D:\python
projects\schoolproject\Fairy.csv",index_col=0))

    elif inp=="C":

        ascdsc()

    elif inp=="D":

        Dinput=input("enter T for seeing top racords/B for
bottom records=")

        if Dinput=="T":

            print("you have selected to see top records")

```



```

        DTinput=int(input("enter number of records you want to see="))

        print(fairyfile.head(DTinput))

    elif Dinput=="B":

        print("you have selected to see bottom records")

        DBinput=int(input("enter number of records you want to see="))

        print(fairyfile.tail(DBinput))

    else:

        print("enter valid input")

elif inp=="E":

    print("we made copy of csv file and it is printed")

    fairynew=fairyfile.to_csv("D:\python projects\schoolproject\Fairynew.csv")

    print(pd.read_csv("D:\python projects\schoolproject\Fairynew.csv"))

elif inp=="F":

    df=pd.read_csv("D:\python
projects\schoolproject\Fairy.csv",usecols=['Name','Total'],index_col=0)

    print(df)

elif inp=="1":

    Fairyline()

elif inp=="2":

    Fairybar()

elif inp=="3":

    Fairypie()

elif inp=="4":

    Fairyscatter()

elif inp=="5":

    Fairyhistogram()

else:

    print("please enter input from above given
options")

# call for taking input for first time

Fairyif()

inp2=input("want to do it again(yes/no)=")

```



```

while inp2=="yes":
    Fairyif()
    inp2=input("want to do it again(yes/no)=")
    if inp2 == "no":
        break
    else:
        print("please enter proper input")

#-----
#fighting function
def Fighting():
    print ("Reading Data from File in Different way")
    print ("A: Read csv file")
    print ("B: Reading file without index")
    print("    ")
    print ("Apply Data Manipulation in the records of csv File")
    print ("C: Sorting the data as per your choice")
    print ("D: Read Top and Bottom Records file as per requirement")
    print ("E: Make the copy of csv file")
    print ("F: Read the Specific column")
    print("    ")
    print ("Data Visualization")
    print ("1: Line Chart")
    print ("2: Bar Plot")
    print ("3: Pie chart")
    print ("4: Scatter chart")
    print("5: Histogram")

    fightingfile=pd.read_csv("D:\python
projects\schoolproject\Fighting.csv")

    f=fightingfile

```



```

def Fightingif():
    inp=input("please enter your  from above given options=")
    if inp=="A":
        print(fightingfile)
    elif inp=="B":
        print(pd.read_csv("D:\python
projects\schoolproject\Fighting.csv",index_col=0))
    elif inp=="C":
        ascdsc()
    elif inp=="D":
        Dinput=input("enter T for seeing top racords/B for bottom records=")
        if Dinput=="T":
            print("you have selected to see top records")
            DTinput=int(input("enter number of records you want to see="))
            print(fightingfile.head(DTinput))
        elif Dinput=="B":
            print("you have selected to see bottom records")
            DBinput=int(input("enter number of records you want to see="))
            print(fightingfile.tail(DBinput))
        else:
            print("enter valid input")
    elif inp=="E":
        print("we made copy of csv file and it is printed")
        fightingnew=fightingfile.to_csv("D:\python
projects\schoolproject\Fightingnew.csv")
        print(pd.read_csv("D:\python projects\schoolproject\Fightingnew.csv"))
    elif inp=="F":

```



```

df=pd.read_csv("D:\python
projects\schoolproject\Fighting.csv",usecols=['Name','Total'],index_col=0)

print(df)

elif inp=="1":

    Fightingline()

elif inp=="2":

    Fightingbar()

elif inp=="3":

    Fightingpie()

elif inp=="4":

    Fightingscatter()

elif inp=="5":

    Fightinghistogram()

else:

    print("please enter input from above given options")


Fightingif()

# call for taking input for first time

inp2=input("want to do it again(yes/no)=")

while inp2=="yes":

    Fightingif()

    inp2=input("want to do it again(yes/no)=")

    if inp2 == "no":

        break


#####      functions      #####


#call1 after info

proceed()


#pie chart

```



```

print("pie chart.    please close pie chart.")

ptype=["Bug","Dark","Dragon","Electric","Fairy","Fighting","Fire","Ghost","Grass","Ground","ice","Normal","poison","Psychic","Rock","water"]

pnumber=[70,32,32,44,17,27,52,32,70,33,24,98,28,37,44,112]

plt.title("Pokemon distribution in types")

plt.pie(pnumber,labels=ptype)

plt.show()

print("    ")

#call2 after pie chart

proceed()


#show file

print("type 1 to see file with index")

print("type any other number to see file without index")

print("    ")

i=int(input("your input="))

if i ==1:

    print("showing file with index")

    mainfile=pd.read_csv("D:\python projects\schoollproject\Pokemon.csv")

    print(mainfile)

else:

    print("showing file without index")

    mainfile=pd.read_csv("D:\python projects\schoollproject\Pokemon.csv",index_col=0)

    print(mainfile)


#call3 after mainfile with index

proceed()

print("    ")

print("there are many types but we will go with six types in this project.")

print("    ")

```




```

#Flowchart
print("flowchart image")
im=Image.open("D:\python projects\schoolproject\Pokemon-flowchart.png")
im.show()

#call4 after flowchart
proceed()

#-----

#bug type
print("BUG TYPE".center(40,"-"))

print("Bug-type Pokémon are generally arthropod-like Pokémon, mostly insectoid and a few arachnids. These Pokémon commonly evolve at low levels and as such are ideal for the early stages of the games. Bug-type moves involve use of the bugs' body parts. Bug Pokémon are generally regarded as weak due to the first generation's Bug types having low stats and the first generation's Bug type moves being weak, but some can be quite powerful, like Heracross. Bug types are weak against Fire, Flying, and Rock types, yet are strong against Grass, Psychic, and Dark types.")

print("    ")

#input for bug type
bugi=input("enter if you want to see graphs of Bug type(yes/no)=")

if bugi=="yes":
    Bug()
else:
    print(nxtype)
print("    ")
#end of bug type
if bugi == "yes":
    print("Thanks for viewing our project")
    sys.exit("program terminated")
else:

```



```

    print("here starts another type")

print("    ")

print("    ")

#-----

#dark type

print("DARK TYPE".center(40,"-"))

print("In the Japanese version, this type is called Evil. According to most Pokédex
information, Dark type Pokémon tend to have bad reputations and an evil nature about
them. Dark-type Pokémon are known for using sinister moves such as biting and stealing.
Examples of Dark-type Pokémon include Tyranitar, Absol, Mightyena, Umbreon, Houndoom,
Sneasel and especially Murkrow for reasonable reasons. Certain species of Pokémon
classified as Dark seem to be misunderstood, such as Absol, who has gathered a reputation
of bad luck, always appearing at human towns when a natural disaster is about to happen,
when really, it tries to warn the humans. Dark type Pokémon are strong against Psychic
and Ghost types, however are weak against Fighting, Bug, and Fairy types.")

print("    ")

#input for dark type

darki=input("enter if you want to see graphs of Dark type(yes/no)=")

if darki=="yes":

    Dark()

else:

    print(nxtype)

print("    ")

#end of Dark type

if darki== "yes":

    print("Thanks for viewing our project")

    sys.exit("program terminated")

else:

    print("here starts another type")

print("    ")

print("    ")

#-----

```



```
#Dragon type
```

```
print("DRAGON TYPE".center(40,"-"))
```

```
print("Dragon-type Pokémon are, quite simply, dragons. Their moves involve the use of claws and breath. They are one of only two types, the other being Ghost, to be Super Effective against its own type. Not all dragon-like Pokémon are Dragon-type Pokémon; for example, Charizard is a Fire/Flying type, Gyarados is a Water/Flying type, Lapras is a Water/Ice type, Aerodactyl is a Rock/Flying type, Steelix is a Steel/Ground type, Tyranitar is a Rock/Dark type, Sceptile is a pure Grass type, Aggron is a Steel/Rock type, and Milotic is pure Water type, but a select few are in the Dragon breeding group (Charizard, Milotic, Gyarados, and Sceptile), and some can Mega Evolve into a Dragon-type (Charizard, Ampharos, and Sceptile).")
```

```
print("    ")
```

```
#input for dragon type
```

```
dragoni=input("enter if you want to see graphs of Dragon type(yes/no)=")
```

```
if dragoni=="yes":
```

```
    Dragon()
```

```
else:
```

```
    print(nxtype)
```

```
print("    ")
```

```
#end of Dragon type
```

```
if dragoni == "yes":
```

```
    print("Thanks for viewing our project")
```

```
    sys.exit("program terminated")
```

```
else:
```

```
    print("here starts another type")
```

```
print("    ")
```

```
print("    ")
```

```
#-----
```

```
#electric type
```

```
print("Electric TYPE".center(40,"-"))
```

```
print("Electric-type Pokémon have electricity-oriented powers. Electric types often have a high speed status, a respectable Special Attack, and a decent Special Defense. Electric types are also noted for very few weaknesses-Although the 1 weakness
```



they have is possibly the 2nd most common. Some examples include Pikachu, Elekid, Raichu, Manetric, Ampharos, Raikou, Plusle, Minun, and Electrode.")

```
print("Many Electric-type attacks have a chance of causing Paralysis, a status effect which severely reduces the affected Pokémon's Speed, and means a 25% chance of being unable to move each turn. Some Electric-type moves that Pokémon can learn are ThunderShock, Shock Wave, and Spark.")
```

```
print("    ")
```

```
#input for electric type
```

```
electrici=input("enter if you want to see graphs of Electric type(yes/no)=")
```

```
if electrici=="yes":
```

```
    Electric()
```

```
else:
```

```
    print(nxtttype)
```

```
print("    ")
```

```
#end of Electric type
```

```
if electrici == "yes":
```

```
    print("Thanks for viewing our project")
```

```
    sys.exit("program terminated")
```

```
else:
```

```
    print("here starts another type")
```

```
print("    ")
```

```
print("    ")
```

```
#-----
```

```
#Fairy type
```

```
print("FAIRY TYPE".center(40,"-"))
```

print("Fairy-type Pokémon have feminine appearances, but four of them have more of a chance of being male than female despite their feminine appearances. They have magical powers and their attacks are super effective against the Dark, Dragon, and Fighting types. Although it is said that dragons were unaffected by magic, the Fairy type is immune to their type and not the other way around. Examples of Fairy-type Pokémon include Flabébé, Togepi, Swirlix, Diancie, Florges, Marill, Cottonee, Sylveon, Mawile, Snubbull, and Xerneas.")

```
print("    ")
```



```

#input for fairy type

fairyi=input("enter if you want to see graphs of Fairy type(yes/no)=")

if fairyi=="yes":
    Fairy()
else:
    print(nxtttype)
print("    ")
#end of Fairy type

if fairyi == "yes":
    print("Thanks for viewing our project")
    sys.exit("program terminated")
else:
    print("here starts another type")
print("    ")
print("    ")
#-----

#Fighting type

print("FIGHTING TYPE".center(40,"-"))

print("Fighting-type Pokémon learn specifically labelled fighting melee attacks, such
as punches and kicks. Examples include Hitmonlee, Hitmonchan, Hitmontop, Lucario,
Hariyama, Mankey, and Machoke. Some Fighting-type moves that Pokémon can learn are
Seismic Toss, Hi Jump Kick, and DynamicPunch.")

print("    ")

#input for fighting type

fightingi=input("enter if you want to see graphs of Fighting type(yes/no)=")

if fightingi == "yes":
    Fighting()
else:
    print("project completed")
print("    ")
#end of fighting type

```



```
print("Thanks for viewing our project")
```

```
sys.exit("program ended")
```

```
#-----
```

