

```

In [ ]: from collections import defaultdict
        from tkinter import *
        queue = []
        t = ' '
        def maps ():
            return ''

pokeCity = ['johto', 'kanto', 'fiore', 'obliva', 'sinnoh', 'ransei', 'almia', 'hoenn', 'orre', 'kalos', 'unova']
linkCity = defaultdict(maps)
linkCity ['johto'] = ['kanto']
linkCity ['kanto'] = ['johto', 'obliva', 'fiore']
linkCity ['fiore'] = ['kanto', 'sinnoh', 'ransei']
linkCity ['ransei'] = ['fiore', 'sinnoh', 'orre']
linkCity ['orre'] = ['ransei', 'hoenn', 'unova']
linkCity ['unova'] = ['orre', 'hoenn']
linkCity ['hoenn'] = ['orre', 'unova', 'kalos']
linkCity ['kalos'] = ['hoenn', 'almia']
linkCity ['almia'] = ['kalos', 'obliva']
linkCity ['obliva'] = ['almia', 'kanto', 'sinnoh']
linkCity ['sinnoh'] = ['obliva', 'fiore', 'ransei']

def linkcity():
    city = False
    while city == False:
        firstcity = input("Please enter the first city's name > ")
        if (firstcity in pokeCity):
            print("Ok, wait a moment")
            city = True
        else:
            print("Please enter it again")

    city2 = False
    while city2 == False:
        endcity = input("Please enter the last city's name > ")
        if(endcity in pokeCity):

```

```

        print("Ok, wait a moment")
        city2 = True
    else:
        print("Please enter it again")

    if(city == True and city2 == True):
        if (endcity in linkCity[firstcity]):
            print("it's linked !")
            print(linkCity[firstcity])
        elif (endcity not in linkCity[firstcity]):
            print("Linking ...")
            linkCity[firstcity].append(endcity)
            print(linkCity[firstcity])
            linkCity[endcity].append(firstcity)
            print(linkCity[endcity])

    progstart()

def delinkcity():
    citydelink = False
    while citydelink == False:
        delink1 = input("Please enter the first city you want to delink
> ")
        if(delink1 in pokeCity):
            print("Ok wait a moment")
            citydelink = True
        else :
            print("Please enter it again")

    citydelink2 = False
    while citydelink2 == False:
        delink2 = input("Please enter the other city you want to delink
with the first one > ")
        if(delink2 in pokeCity):
            print("Ok wait a moment")
            citydelink2 = True
        else:
            print("Please enter it again")

```

```

    if(citydelink == True and citydelink2 == True):
        if(delink2 in linkCity[delink1]):
            print("Delinking ....")
            linkCity[delink1].remove(delink2)
            print(linkCity[delink1])
            linkCity[delink2].remove(delink1)
            print(linkCity[delink2])
        elif(delink2 not in linkCity[delink1]):
            print("it's not linked")

    progstart()

def searchlink():
    search = False
    while search == False:
        searching = input("Please enter the city you are looking for > ")
    )
        if searching in pokeCity:
            print("Showing ...")
            print(linkCity[searching])
            search = True
        else :
            print("please try again")
            search = False
            searchlink()

    progstart()

def BFSEARCH():
    BFS = input ("Please enter the path you are looking for > ")
    path = []
    queue = [BFS]
    if BFS in pokeCity:
        while queue:
            element = queue.pop(0)
            if element not in path:
                path.append(element)
                queue.extend(linkCity[element])
        print (path)

```

```

else:
    print("Please enter the correct city")
    BFSEARCH()
    progstart()

#main main main
def createGUI():
    def linkcity_GUI():
        city = False
        while city == False:
            firstcity = tkvar.get()
            if (firstcity in pokeCity):
                city = True
            else:
                t=("Please enter it again")

        city2 = False
        while city2 == False:
            endcity = tkvar2.get()
            if(endcity in pokeCity):
                city2 = True
            else:
                t = ("Please enter it again")

        if(city == True and city2 == True):
            if (endcity in linkCity[firstcity]):

                t = "it's linked already !"
            elif (endcity not in linkCity[firstcity]):

                linkCity[firstcity].append(endcity)
                linkCity[endcity].append(firstcity)
                t = firstcity + " and " + endcity + " is linked."
            output.delete(0.0, END)
            output.insert(END, t)
    def delinkcity_GUI():
        citydelink = False
        while citydelink == False:

```

```

delink1 = tkvar.get()
if(delink1 in pokeCity):
    citydelink = True
else :
    t = "Please enter it again"

citydelink2 = False
while citydelink2 == False:
    delink2 = tkvar2.get()
    if(delink2 in pokeCity):
        citydelink2 = True
    else:
        t ="Please enter it again"

if(citydelink == True and citydelink2 == True):
    if(delink2 in linkCity[delink1]):
        linkCity[delink1].remove(delink2)
        linkCity[delink2].remove(delink1)
        t = delink1 + " and " + delink2 + " is no more longer l
inked together !"
    elif(delink2 not in linkCity[delink1]):
        t="it's not linked"
        output.delete(0.0, END)
        output.insert(END, t)
def searchlink_GUI():
    search = False
    while search == False:
        searching = str(textentry3.get())
        if searching in pokeCity:
            t ="Neighbour city of " + searching + " > " + ', '.join
(linkCity[searching])
            search = True
        else :
            t ="please try again"
            search = True
        output.delete(0.0, END)
        output.insert(END, t)
def BFSEARCHGUI():
    BFS = str(textentry3.get())

```

```

path = []
queue = [BFS]
if BFS in pokeCity:
    while queue:
        element = queue.pop(0)
        if element not in path:
            path.append(element)
            queue.extend(linkCity[element])
    t = ' -> '.join(path)
else :
    t = "please enter the correct city"
output.delete(0.0, END)
output.insert(END,t)

```

```

window = Tk()
window.title("Welcome to Poke City")

```

```

window.configure(background ="yellow")
#photo of pokecity

```

```

Label (window, bg="yellow").grid(row=0,column=0)
emptyLabel = Label(window, bg="yellow")
emptyLabel.grid(row=4,column=5, sticky=E)

```

```

#create label

```

```

L1 = Label(window, text="HEY THIS IS POKE CITY SYSTEM", bg="yellow"
, fg="black", font="none 12 bold underline")
L1.grid(row = 1,column=0, columnspan=2)

```

```

L2 = Label(window, text = "Enter First city", bg="yellow", fg="black", font ="none 12 bold")
L2.grid(row=3,column=0,columnspan=1)

```

```

L3 = Label(window, text = "Enter Second city", bg="yellow", fg="black", font ="none 12 bold")
L3.grid(row=4,column=0,columnspan=1)

```

```

L4 = Label(window, text = "Enter here to search", bg="yellow", fg=
"black", font="none 12 bold")
L4.grid(row = 6,column=0,columnspan=1, sticky=W)
L6 = Label(window, text = "> TO LINK OR DELINK <", bg = "yellow", fg
="black", font="none 12 " )
L6.grid(row=2, column=0, columnspan=1)
L7 = Label(window, text = "> TO SEARCH <", bg = "yellow", fg="black"
, font="none 12" )
L7.grid(row=5, column=0, columnspan=1)

#create tkinker
tkvar = StringVar(window)
tkvar2 = StringVar(window)
#dropdown list
choices ={'johto','kanto','fiore','obliva','sinnoh','ransei','almi
a','hoenn','orre','kalos','unova'}
tkvar.set('johto')
tkvar2.set('johto')
#create text box
textentry1 = OptionMenu(window, tkvar, *choices)
textentry1.grid(row=3,column=1, sticky=W)
textentry2 = OptionMenu(window, tkvar2, *choices)
textentry2.grid(row=4,column=1, sticky=W)
textentry3 = Entry(window, width=12, bg="white")
textentry3.grid(row=6, column=1, sticky=W)
#submit button
B1 = Button(window, text="BFSEARCH", width=10, borderwidth=2, comma
nd=BFSEARCHGUI)
B1.grid(row=7,column=2, columnspan=2, sticky=E)
B2 = Button(window, text="LINK", width = 10, borderwidth=2, command
=linkcity_GUI)
B2.grid(row=3, column =2, columnspan=2,sticky=E)
B3 = Button(window, text="DELINK", width =10, borderwidth=2, comman
d=delinkcity_GUI)
B3.grid(row=4, column=2, columnspan=2, sticky=E)
B4 = Button(window, text="NEIGHBOUR", width=10, borderwidth=2, comm
and=searchlink_GUI)
B4.grid(row=6,column=2,columnspan=2,sticky=E)
#another label

```

```

    L4 = Label (window, text="\nOUTPUT  ", bg="yellow", fg="black", font="none 12 bold underline")
    L4.grid(row=7,column=0, sticky=W)
    #another text box
    output = Text(window, width=50,height=4, background="white")
    output.grid(row=8,column=0, columnspan = 3)

    #exit toggle
    L5 = Label(window, text="press to exit", bg="yellow", fg="black", font="none 12 bold")
    L5.grid(row=9,column=1,sticky=E)
    #function for exit
    def close_window():
        window.destroy()
        exit()
    #exit button
    B6 = Button(window, text="EXIT", width=10, borderwidth=2, command=close_window)
    B6.grid(row=9,column=2,sticky=E)

    window.mainloop()

def progstart():
    #print("Please choose your option > ")
    menu = False
    while menu == False:

```



```

        menuinput = int(input("1. Linking \n2. Delinking \n3. Search ne
ighbour city \n4. Find the path to city \n5. enter 5 for GUI \n6. enter
0 to exit \nPlease choose your option > "))
        if(menuinput == 1):
            linkcity()
            menu = True
        elif(menuinput == 2):
            delinkcity()
            menu = True
        elif(menuinput == 3):
            searchlink()
            menu = True
        elif(menuinput == 4):
            BFSEARCH()
            menu = True
        elif(menuinput == 5):
            createGUI()
            progstart()
            menu = True
        elif(menuinput == 0):
            break
        else :
            print("please choose the number 0-5")
            progstart()

progstart()

```

```

1. Linking
2. Delinking
3. Search neighbour city
4. Find the path to city
5. enter 5 for GUI
6. enter 0 to exit
Please choose your option > 5

```

In [ ]: