### Line arrow Slight curveFirst we created notepad application by using python tkinter module, It looks similar to default application

from tkinter import \*  
from tkinter import Menu  
from tkinter.filedialog import askopenfilename, asksaveasfilename  
import os  
global text  
global master

def runpad ():  
 if True:  
 global master  
 master = Tk()  
 master.title("Notepad")  
 master.geometry("440x720")  
 # type area  
  
 bgcol = 'white'  
  
 fgcol = 'black'  
 global text  
 text = Text(master, font="lucida 13", bg=bgcol, fg=fgcol)  
 global file  
 file = None  
 text.pack(expand =True ,fill =BOTH)  
 # menubar  
  
 menu\_bar = Menu(master)  
 file\_menu = Menu(menu\_bar, tearoff=0)  
 # file  
 # new file  
 file\_menu.add\_command(label="New", command=newFile)  
 # open file  
 file\_menu.add\_command(label="Open", command=openFile)  
  
 file\_menu.add\_command(label="Save", command=saveFile)  
 file\_menu.add\_command(label="change theme", command=change\_th)  
 file\_menu.add\_command(label="Convert to Handwriting", command=saveFile1)  
 file\_menu.add\_separator()  
 file\_menu.add\_command(label="Exit", command=quitApp)  
 menu\_bar.add\_cascade(label="File", menu=file\_menu)  
 # edit menu  
 edit\_menu = Menu(menu\_bar, tearoff=0)  
 edit\_menu.add\_command(label="cut", command=cut)  
 edit\_menu.add\_command(label="copy", command=copy)  
 edit\_menu.add\_command(label="paste", command=paste)  
 menu\_bar.add\_cascade(label="Edit", menu=edit\_menu)  
 master.config(menu=menu\_bar)  
 Scroll = Scrollbar(text)  
 Scroll.pack(side=RIGHT, fill=Y)  
 Scroll.config(command=text.yview)  
 text.config(yscrollcommand=Scroll.set)  
 master.mainloop()  
runpad()

### Line arrow Slight curveThen we will add function mentions in above code such as newFile, cut, copy, past, etc…

def newFile():  
 global file  
 master.title("Untitled ")  
 file =None  
 text.delete(1.0,END)  
def openFile():  
 global file  
 file = askopenfilename(defaultextension=".txt",filetypes=[("All Files", "\*.\*"),("Text Documents", "\*.txt")])  
 if file == "":  
 file = None  
 else:  
 master.title(os.path.basename(file) + " - Notepad")  
 text.delete(1.0, END)  
 f = open(file, "r")  
 text.insert(1.0, f.read())  
 f.close()

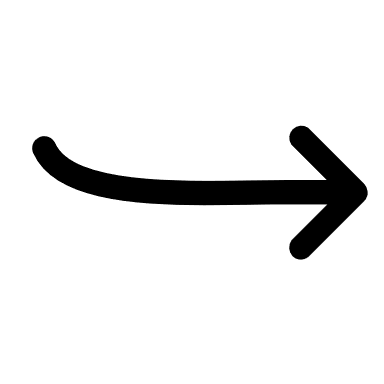
def saveFile():  
 global file  
 if file == None:  
 file = asksaveasfilename(initialfile='Untitled.txt', defaultextension=".txt",  
 filetypes=[("All Files", "\*.\*"),  
 ("Text Documents", "\*.txt")])  
 if file == "":  
 file = None  
  
 else:  
 # Save as a new file  
 f = open(file, "w")  
 f.write(text.get(1.0, END))  
 f.close()  
  
 master.title(os.path.basename(file) + " - Notepad")  
 print("File Saved")  
 else:  
 # Save the file  
 f = open(file, "w")  
 f.write(text.get(1.0, END))  
 f.close()  
  
def cut():  
 text.event\_generate(("<<Cut>>"))  
def copy():  
 text.event\_generate(("<<Copy>>"))  
def paste():  
 text.event\_generate(("<<Paste>>"))  
def quitApp():  
 master.destroy()

now add function like change theme and convert to Handwriting

### Line arrow Slight curveFor change theme I just re run “runpad” function again with dark theme

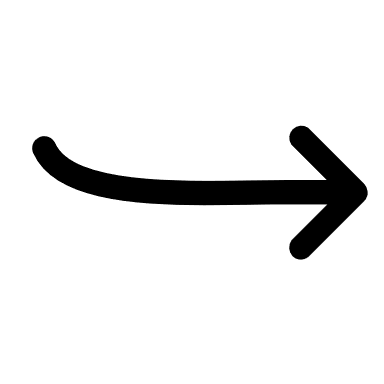
def change\_th():  
# do know what to so if i change "text " to globle othrt function cant use it and if i dont thhen i cant change colour  
  
#methord 3  
 if True:  
 global master  
 master = Tk()  
 master.title("Notepad")  
 master.geometry("440x720")  
 # type area  
  
 bgcol = 'black'  
  
 fgcol = 'white'  
 global text  
 text = Text(master, font="lucida 13", bg=bgcol, fg=fgcol)  
 global file  
 file = None  
 text.pack(expand=True, fill=BOTH)  
 # menubar  
  
 menu\_bar = Menu(master)  
 file\_menu = Menu(menu\_bar, tearoff=0)  
 # file  
 # new file  
 file\_menu.add\_command(label="New", command=newFile)  
 # open file  
 file\_menu.add\_command(label="Open", command=openFile)  
  
 file\_menu.add\_command(label="Save", command=saveFile, bgcol='black', fgcol='white')  
 file\_menu.add\_command(label="change theme", command=change\_th)  
 file\_menu.add\_command(label="convet to handwriting", command=saveFile1)  
 file\_menu.add\_separator()  
 file\_menu.add\_command(label="Exit", command=quitApp)  
 menu\_bar.add\_cascade(label="File", menu=file\_menu)  
 # edit menu  
 edit\_menu = Menu(menu\_bar, tearoff=0)  
 edit\_menu.add\_command(label="cut", command=cut)  
 edit\_menu.add\_command(label="copy", command=copy)  
 edit\_menu.add\_command(label="paste", command=paste)  
 menu\_bar.add\_cascade(label="Edit", menu=edit\_menu)  
 master.config(menu=menu\_bar)  
 Scroll = Scrollbar(text)  
 Scroll.pack(side=RIGHT, fill=Y)  
 Scroll.config(command=text.yview)  
 text.config(yscrollcommand=Scroll.set)  
 master.mainloop()

### ignore my comment in code that’s only for me 😊

For convert to handwriting create new python file “algo.py”

* Import image ,FPDF
* Add function

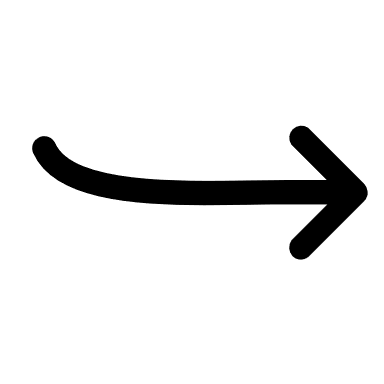
def write(char):  
 global gap, \_  
 if char == '\n':   
 gap = 25  
 \_ += 150  
  
 elif char == '\t':   
 \_ += 150  
 gap = 0  
  
 else:  
 char.lower()  
 cases = Image.open("%s.png" % char)  
 BG.paste(cases, (gap, \_))  
 size = cases.width  
 gap += size  
 del cases  
  
  
def letterwrite(word):  
 global gap, \_  
 if gap > sizeOfSheet - 50\*(len(word)):  
 gap = 25  
 \_ += 20  
 for letter in word:   
 if letter in allowedChars:  
 if letter.islower():  
 pass  
 elif letter.isupper():  
 letter = letter.lower()  
 letter += 'upper'   
 elif letter == '.':  
 letter = "fullstop"  
 elif letter == '!':  
 letter = 'exclamation'  
 elif letter == '?':  
 letter = 'question'  
 elif letter == ',':  
 letter = 'comma'  
 elif letter == '(':  
 letter = 'braketop'  
 elif letter == ')':  
 letter = 'braketcl'  
 elif letter == '-':  
 letter = 'hiphen'  
 elif letter == '\n':  
 letter = '\n'  
  
 write(letter)  
  
  
def word(Input):  
 wordlist = Input.split(' ')  
 for i in wordlist:  
 letterwrite(i)  
 write('space')



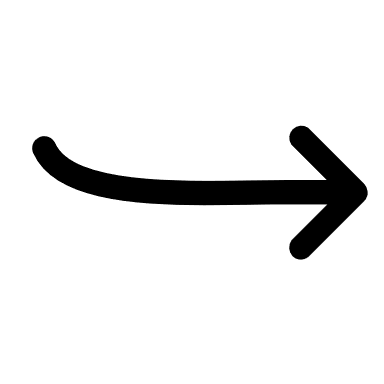
Then run main loop

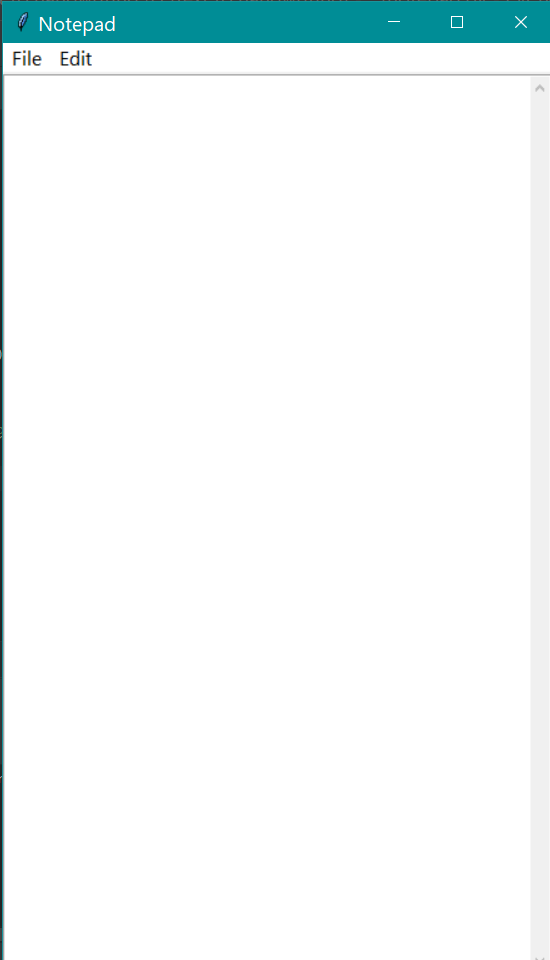
tru = True  
if tru:  
 try:  
 with open('input.txt', 'r') as file:  
 data = file.read().replace('\n', '\n')  
 l = len(data)  
 nn = len(data)//600  
 chunks, chunk\_size = len(data), len(data)//(nn+1)  
 p = [data[i:i+chunk\_size] for i in range(0, chunks, chunk\_size) ]  
   
 for i in range(0,len(p)):  
 word(p[i])  
 write('\n')  
 BG.save('%doutt.png'%i)  
 BG1= Image.open("bg.png")  
 BG=BG1  
 gap = 0  
 \_ =0  
 except ValueError as E:  
 print("{}\nTry again".format(E))  
 tru = False  
from PIL import Image  
from fpdf import FPDF  
imagelist=[]  
for i in range(0,2):  
 imagelist.append('%doutt.png'%i)  
cover = Image.open(imagelist[0])  
width, height = cover.size  
  
pdf = FPDF(unit = "pt", format = [width, height])  
for i in range(0,len(imagelist)):  
 pdf.add\_page()  
 pdf.image(imagelist[i], 0, 0)  
pdf.output("output.pdf", "F")

As you have notice that it read only “input.txt” file because I am lazy to add file chousing function in code

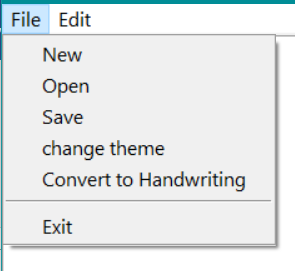
Now I have imported this algo.py to notepad.py as an function

def algo():  
 # import the time module  
 global file, p  
 if file == None:  
 file = asksaveasfilename(initialfile='input.txt', defaultextension=".txt",  
 filetypes=[("All Files", "\*.\*"),  
 ("Text Documents", "\*.txt")])  
 if file == "":  
 file = None  
  
 else:  
 # Save as a new file  
 f = open(file, "w")  
 f.write(text.get(1.0, END))  
 f.close()  
  
 master.title(os.path.basename(file) + " - Notepad")  
 print("File Saved")  
 else:  
 # Save the file  
 f = open(file, "r")  
 f.write(text.get(1.0, END))  
 f.close()  
 import tkinter.messagebox  
 tkinter.messagebox.showinfo('Note', ' It will take some time')  
 from Algo import write  
 from Algo import word  
 from Algo import letterwrite  
 from Algo import BG  
 from Algo import BG1  
 from PIL import Image  
 tru = True  
 if tru:  
 try:  
 with open('input.txt', 'r+') as file:  
 data = file.read().replace('\n', '\n')  
 l = len(data)  
 nn = len(data) // 600  
 chunks, chunk\_size = len(data), len(data) // (nn + 1)  
 p= [data[i:i + chunk\_size] for i in range(0, chunks, chunk\_size)]  
  
 for i in range(0, len(p)):  
 word(p[i])  
 write('\n')  
 BG.save('%doutt.png' % i)  
 BG1 = Image.open("bg.png")  
 BG = BG1  
 gap = 0  
 \_ = 0  
  
 except ValueError as E:  
 print("{}\nTry again".format(E))  
  
  
 tkinter.messagebox.showinfo('Process Done ', ' Now you can Check the File')

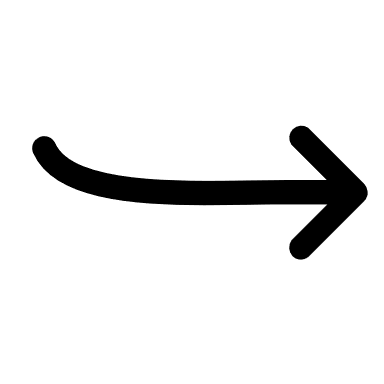
Run notepad.py

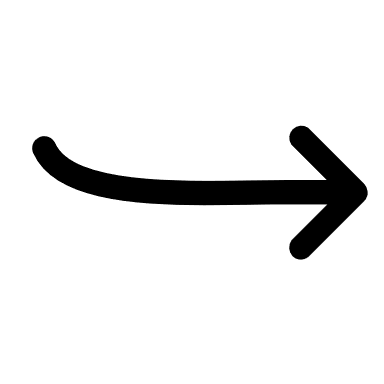
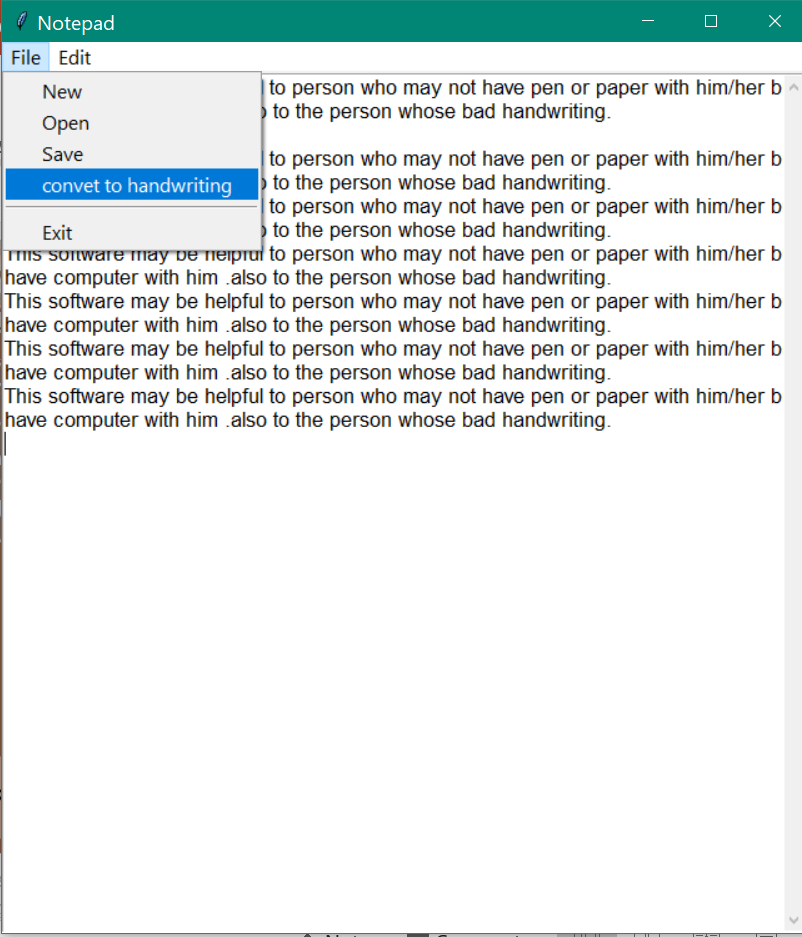


We can write and edit text in this just like normal notepad.

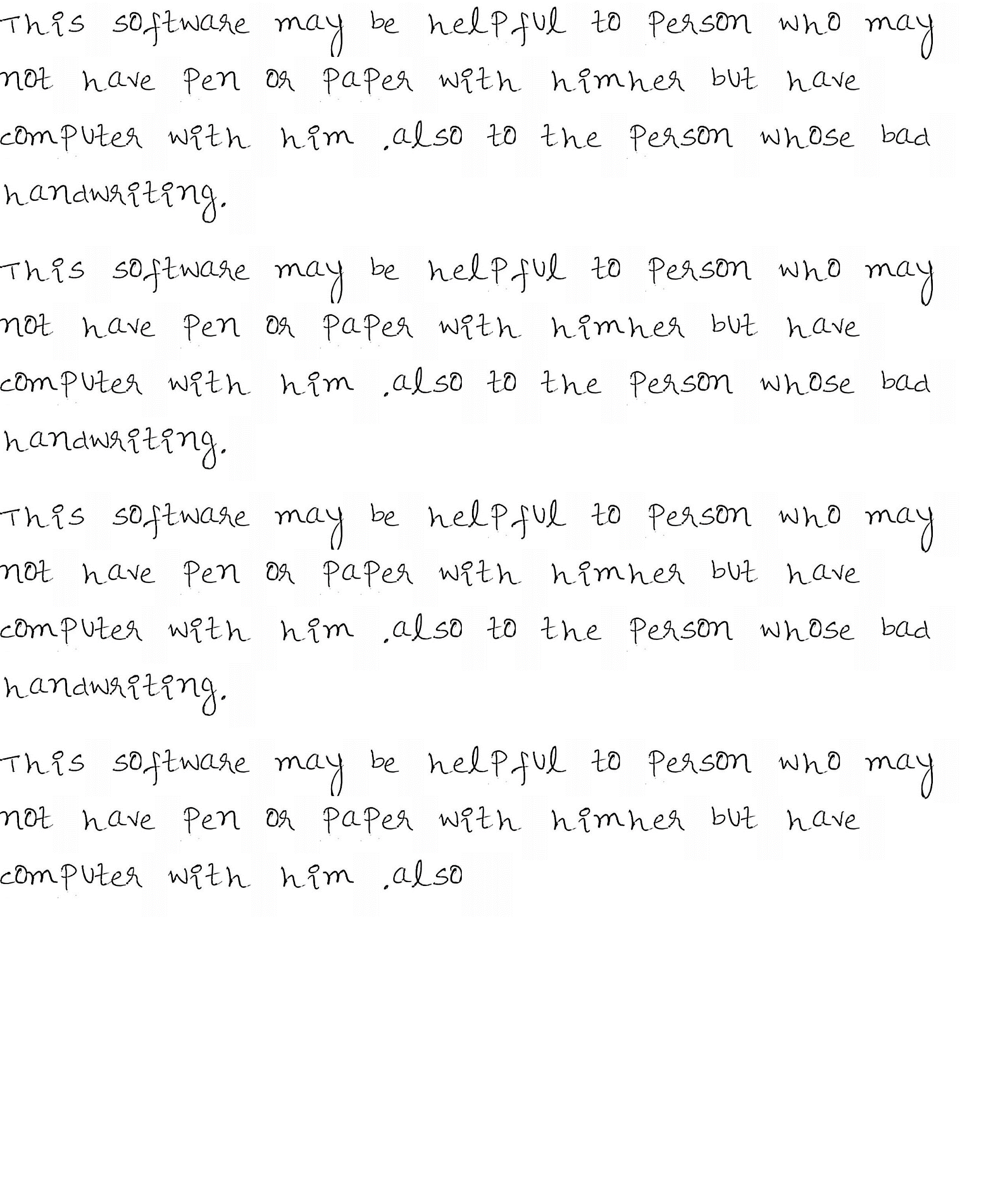


Layout may look like this .

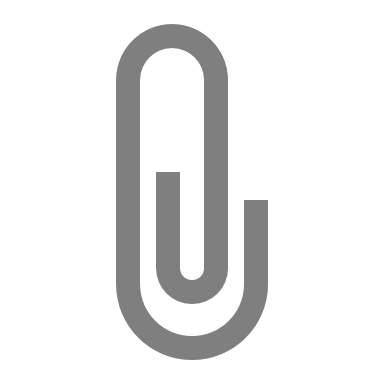
Input

### output in .png format



File attachment

[](read_me.docx)

[Read\_me](read_me.docx)

[notepad\_setup.exe](notepad_setup.exe)