```
'''PROJECT TITLE: PASSWORD GENERATOR
   CODE LANGUAGE: PYTHON 3.7.0a3(32-bit)
   FILE NAME: passwordGenerator.py'''

'''SOURCE CODE:'''
import random
from tkinter import *
from tkinter.ttk import *
import time
import datetime

def qwe(len,ch):
    def copy():
        import pyperclip
        c=""
        s=(lis.get(0,END))
        for i in s:
            c+=i+"\n"
        pyperclip.copy(c.strip())
    def Reset():
        lis.delete(0,END)
        E1.delete(0,END)
        entry5.delete(0,END)
        entry5.insert(0,"Enter the count")
    def msg(s):
        entry5.insert(END, s)
    def reqpw_except(cc):
        reqPWs=cc
        try:
            aa=int(reqPWs)
            if(aa==0):
                si="Enter count greater than 0"
                return msg(si)
            elif aa<0:
                si="Error:Negative value!"
                return msg(si)
            else:return aa
        except ValueError:
            si="Error:Invalid Input count"
            return msg(si)
    def qExit():
        root1.destroy()
    def multiple(len,ch):
        entry5.delete(0,END)
        count=E1.get()
        count4=reqpw_except(count)
        password=""
        for i in range(0, count4):
            for j in range(len):
                password+=random.choice(ch)
            password+="\n"
        entry5.insert(0,"Copy your passwords")
        lis.delete(0,END)
        for index,i in enumerate(password.split("\n")):
            lis.insert(index,i)

    # create GUI window
    root1=Tk()
    root1.title("Multiple Password generator")
    fm1 = Frame(root1, width = 20, relief = SUNKEN)
    fm1.pack(side = TOP,padx=10,pady=10)
    fm = Frame(root1, width = 20, relief = SUNKEN)
    fm.pack(side = TOP,padx=10,pady=10)
    fm2 = Frame(root1, width = 20, relief = SUNKEN)
    fm2.pack(side = TOP,padx=10,pady=10)
    fm3 = Frame(root1, width = 20, relief = SUNKEN)
    fm3.pack(side = TOP,padx=10,pady=10)
    fm4 = Frame(root1, width = 20, relief = SUNKEN)
    fm4.pack(side = TOP,padx=10,pady=10)
    password_count = Label(fm1, text=" Password Count: ",font = ('timesnewroman',
    12,'bold'))
    password_count.pack(side=LEFT,padx=2,pady=2)
```

```
72        E1 = Entry(fm1,font = ('timesnewroman', 15))
73        E1.pack(side = LEFT)
74        lblInfo =Label(fm, font=('timesnewroman',12,'bold'),text = " Message:
          ")
75        lblInfo.pack(side=LEFT,padx=2,pady=2)
76        entry5 =Entry(fm,width=35,font = ('timesnewroman', 15, 'bold'))
77        entry5.pack(side=LEFT)
78        entry5.insert(0,"Enter password count")
79        scrollbar=Scrollbar(fm2)
80        scrollbar.pack(side=RIGHT, fill=Y)
81        lis=Listbox(fm2,width=50,height=10,font=('timesnewroman,14,bold'))
82        lis.pack(side=LEFT,padx=5,pady=5,expand=1)
83        lis.config(yscrollcommand=scrollbar.set)
84        scrollbar.config(command=lis.yview)
85        generate_button = Button(fm3,text="Generate",command=lambda:multiple(len,ch))
86        generate_button.pack(side=LEFT)
87        copy_button1 = Button(fm3, text="Copy",command=copy)
88        copy_button1.pack(side=LEFT)
89        reset1= Button(fm3, text="Reset",command=Reset)
90        reset1.pack(side=LEFT)
91        exit1= Button(fm3, text="Exit",command=qExit)
92        exit1.pack(side=LEFT)
93        localtime = time.asctime(time.localtime(time.time()))
94        lblInfo = Label(fm4, font=('timesnewroman',12,'bold'),text =
          localtime)
95        lblInfo.pack(side=BOTTOM)
96        root.mainloop()
97    #single password generator
98    def length_except(a):
99        length=a
100       try:
101           bb=int(length)
102           if(bb==0 or bb==1 or bb==2 or bb==3):
103               si="Select length(4-32)"
104               return msg(si)
105           elif bb>32:
106               si="Enter the Length less than 32"
107               return msg(si)
108           elif bb<0:
109               si="Error:Negative Value Length"
110               return msg(si)
111           else:
112               return bb
113       except ValueError:
114           si="Error:Invalid Input Length"
115           return msg(si)
116   def msg(s):
117       entry1.insert(END, s+"\n                                                                    ")
118   def single(length,ch):
119       password=""
120       for i in range(0, length):
121               password +=random.choice(ch)
122       return password
123
124
125       if value==1:
126           return single(length,ch)
127       else:
128           s="select single/multiple"
129           return msg(s)
130   def check(val,ch,length):
131           if val==1:
132                   return single(length,ch)
133           elif val==2:
134                   if length>=4 and length<=32:
135                           qwe(length,ch)
136           else:
137                   s="select single/multiple"
138                   return msg(s)
139   def calculation():
140       entry.delete(0, END)
141       entry1.delete(0,END)
```

```
    # Get the length of password
    length=var1.get()
    value=var.get()
    length=length_except(length)
    password=""
    num='0123456789'
    SLet='abcdefghijklmnopqrstuvwxyz'
    CLet='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    punc="!\#$%&()*+-/:<=>?@[\\]^_{|}~"
    if v1.get()== 1 and v2.get()== 0 and v3.get()== 0 and v4.get()== 0:
        ch=num
        return check(value,ch,length)
    elif v1.get()== 0 and v2.get()== 0 and v3.get()== 0 and v4.get()== 4:
        ch=punc
        return check(value,ch,length)
    elif v1.get()== 0 and v2.get()== 0 and v3.get()== 3 and v4.get()== 0:
        ch=SLet
        return check(value,ch,length)
    elif v1.get()== 0 and v2.get()== 2 and v3.get()== 0 and v4.get()== 0:
        ch=CLet
        return check(value,ch,length)
    elif v1.get()== 1 and v2.get()== 2 and v3.get()== 3 and v4.get()== 4:
        ch=num+punc+SLet+CLet
        return check(value,ch,length)
    elif v1.get()== 0 and v2.get()== 2 and v3.get()== 3 and v4.get()== 0:
        ch=CLet+SLet
        return check(value,ch,length)
    elif v1.get()== 1 and v2.get()== 0 and v3.get()== 3 and v4.get()== 0:
        ch=num+SLet
        return check(value,ch,length)
    elif v1.get()== 1 and v2.get()== 2 and v3.get()== 0 and v4.get()== 0:
        ch=num+CLet
        return check(value,ch,length)
    elif v1.get()== 0 and v2.get()== 0 and v3.get()== 3 and v4.get()== 4:
        ch=punc+SLet
        return check(value,ch,length)
    elif v1.get()== 0 and v2.get()== 2 and v3.get()== 0 and v4.get()== 4:
        ch=punc+CLet
        return check(value,ch,length)
    elif v1.get()== 1 and v2.get()== 0 and v3.get()== 0 and v4.get()== 4:
        ch=punc+num
        return check(value,ch,length)
    elif v1.get()== 1 and v2.get()== 0 and v3.get()== 3 and v4.get()== 4:
        ch=punc+num+SLet
        return check(value,ch,length)
    elif v1.get()== 1 and v2.get()== 2 and v3.get()== 0 and v4.get()== 4:
        ch=punc+CLet+num
        return check(value,ch,length)
    elif v1.get()== 1 and v2.get()== 2 and v3.get()== 3 and v4.get()== 0:
        ch=num+CLet+SLet
        return check(value,ch,length)
    elif v1.get()== 0 and v2.get()== 2 and v3.get()== 3 and v4.get()== 4:
        ch=punc+SLet+CLet
        return check(value,ch,length)
    else:
        return msg("Tick options")
def generate():
    password = calculation()
    entry.insert(END, password)
    msg("copy your password")
def copy1():
    import pyperclip
    random_password = entry.get()
    pyperclip.copy(random_password)
def qExit():
    root.destroy()
    exit(0)
def Reset():
    v1.set(0)
    v2.set(0)
    v3.set(0)
    v4.set(0)
```

```
    var.set(0)
    var1.set("4")
    entry.delete(0,END)
    entry1.delete(0,END)

# Main Function
# GUI window
root = Tk()
root.geometry("590x370")
root.title("Password Generator")
top = Frame(root, width = 20, relief = SUNKEN)
top.pack(side = TOP,padx=10,pady=10)
tops = Frame(root, width = 20, relief = SUNKEN)
tops.pack(side = TOP,padx=10,pady=10)
radio = Frame(root, width = 20, relief = SUNKEN)
radio.pack(side = TOP,padx=10,pady=10)
f2 = Frame(root, width = 20, height = 20,relief = SUNKEN)
f2.pack(side = TOP,padx=10,pady=10)
fw = Frame(root, width = 20, height = 20,relief = SUNKEN)
fw.pack(side = TOP,padx=10,pady=10)
last = Frame(root, width = 20, height = 20,relief = SUNKEN)
last.pack(side = TOP,padx=10,pady=10)
f5 = Frame(root, width = 20, height = 20,relief = SUNKEN)
f5.pack(side = TOP,padx=10,pady=10)
f4= Frame(root, width = 20, height = 20,relief = SUNKEN)
f4.pack(side = TOP,padx=10,pady=10)

var=IntVar()
v1=IntVar()
v2=IntVar()
v3=IntVar()
v4=IntVar()
var1=StringVar()

Random_password = Label(last, text="Generated Password: ",font = ('timesnewroman',
12,'bold'))
Random_password.pack(side=LEFT,padx=5)
entry =Entry(last,width=30,font = ('timesnewroman', 15, 'bold'))
entry.pack(side=LEFT,padx=2,fill=Y)
c_label = Label(top, text="Length: ",font = ('timesnewroman', 12, 'bold'))
c_label.pack(side=LEFT,padx=5)
combo = Combobox(top, textvariable=var1,font = ('timesnewroman', 15))
combo['values'] = (4,5,6,7,8, 9, 10, 11, 12, 13, 14, 15, 16,
           17, 18, 19, 20, 21, 22, 23, 24, 25,
           26, 27, 28, 29, 30, 31, 32)
combo.current(0)
combo.bind('<<ComboboxSelected>>')
combo.pack(side=LEFT,padx=2)
a=Checkbutton(tops, text="DIGITS",variable=v1,onvalue=1,offvalue=0)
a.pack(side=LEFT,fill=BOTH)
b=Checkbutton(tops, text="UPPER", variable=v2,onvalue=2,offvalue=0)
b.pack(side=LEFT,fill=BOTH)
c=Checkbutton(tops, text="LOWER", variable=v3,onvalue=3,offvalue=0)
c.pack(side=LEFT,fill=BOTH)
d=Checkbutton(tops, text="SYMBOLS", variable=v4,onvalue=4,offvalue=0)
d.pack(side=LEFT,fill=BOTH)
radio_low = Radiobutton(radio,text="SINGLE PASSWORD", variable=var, value=1)
radio_low.pack(side=LEFT,fill=BOTH)
radio_strong = Radiobutton(radio, text="MULTIPLE PASSWORDS", variable=var, value=2)
radio_strong.pack(side=LEFT,fill=BOTH)
generate_button = Button(f2,text="Generate",command=generate)
generate_button.pack(side=LEFT,fill=BOTH)
copy_button = Button(f5, text="Copy",command=copy1)
copy_button.pack(side=LEFT)
copy_button1 = Button(f5, text="Exit",command=qExit)
copy_button1.pack(side=RIGHT)
btnReset = Button(f5,text = "Reset",command = Reset).pack(side=RIGHT)
lblInfo =Label(fw,font=('timesnewroman',12,'bold'),text = " Message:
")
lblInfo.pack(side=LEFT,padx=2,pady=2)
entry1 =Entry(fw,width=36,font = ('timesnewroman', 15, 'bold'))
entry1.pack(side=LEFT)
```

```
284   localtime = time.asctime(time.localtime(time.time()))
285   lblInfo = Label(f4, font=('timesnewroman',12,'bold'),text =
      localtime)
286   lblInfo.pack(side=BOTTOM)
287   root.mainloop()
288
289   '''END CODE'''
290
```