**Password Creator**

***BY-***

***S. Athithian***

***21CS210***

***CSE-A***

Password Generators are a tool to generate passwords based on security guidelines that user sets to create strong and unpredictable password. A strong password must have at least 15 characters consisting of uppercase ascii letters, lowercase ascii letters, numbers, symbols (“! @ # $ % ^ & \* ( ) ”) .

The following source code is written in python. Modules used are *tkinter, random, pyperclip*. Tkinter is used for simple GUI implementation, random is used for creating random choices and pyperclip to copy information onto clipboard.

The basic logic is to:

1. Create an application window

1.1. initialize root as Tk()

1.2. create a window using geometry method of tkinter

1.3. get variables for length and storing password

1.4. generate password using python

1.5. create buttons and set commands as per pre-defined methods

1.6. run infinite mainloop() to run the code through front end (GUI using tkinter)

1. Create a password using python (using random module).
2. Copy and display it in GUI window according to data inputted from user.

The strength of any password is determined by the random algorithms used to generate it and the length of the password. Pseudo-random algorithms are methods such as atmospheric randomness or FIPS181.

A password created using a pseudo random algorithm can be retraced and easily reproduced. The *random module* in python is also a pseudo random as it works on PRNG (pseudorandom number generator). To create a truly random number *SystemRandom* method is used under random module.

The information entropy is calculated for assessing the strength of a random password. It is given by H as

Where L is length of the password

N=number of possible characters

Entropy(H) is measured in bits. A strong password capable of securing financial information has an entropy in the range of 60-127 bits.

Using the current source code for a standard password of length of 15 characters we can get an entropy of 92.84736838320026 bits which is very secure.

If we only use random from random module it will cycle same values, can be reproduced and the randomness is not truly random.

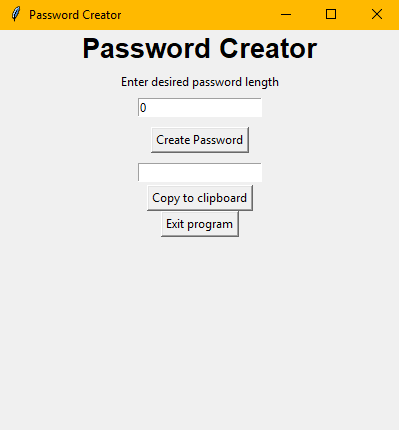
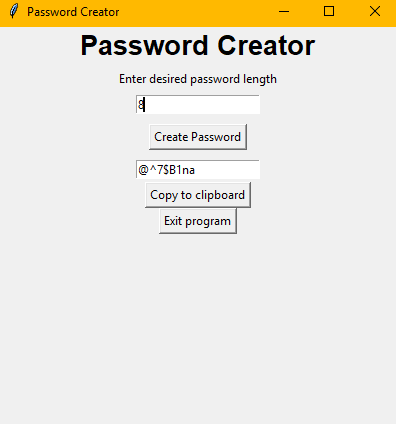
Whereas using SystemRandom() method from random module as random.SystemRandom() generates truly random generations. Thus SystemRandom is used under random.

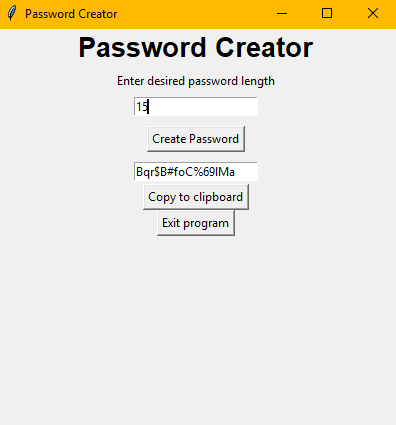
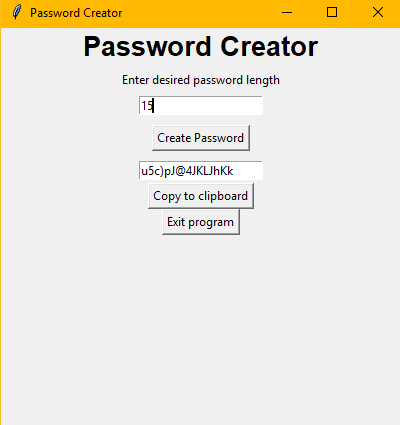
***CODE:***

(WRITTEN IN PYCHARM USING PYTHON 3.9)

from tkinter import \*  
import pyperclip**,**random  
rm=random.SystemRandom()  
root=Tk()  
root.title("Password Creator")  
root.geometry("400x400")  
s=StringVar()  
len=IntVar()  
len.set(**0**)  
def pwd():  
 l1=['a'**,** 'b'**,** 'c'**,** 'd'**,** 'e'**,** 'f'**,** 'g'**,** 'h'**,** 'i'**,** 'j'**,** 'k'**,** 'l'**,** 'm'**,** 'n'**,** 'o'**,** 'p'**,** 'q'**,** 'r'**,** 's'**,** 't'**,** 'u'**,** 'v'**,** 'w'**,** 'x'**,** 'y'**,** 'z'**,** 'A'**,** 'B'**,** 'C'**,** 'D'**,** 'E'**,** 'F'**,** 'G'**,** 'H'**,** 'I'**,** 'J'**,** 'K'**,** 'L'**,** 'M'**,** 'N'**,** 'O'**,** 'P'**,** 'Q'**,** 'R'**,** 'S'**,** 'T'**,** 'U'**,** 'V'**,** 'W'**,** 'X'**,** 'Y'**,** 'Z'**,** '1'**,** '2'**,** '3'**,** '4'**,** '5'**,** '6'**,** '7'**,** '8'**,** '9'**,** '0'**,** ' '**,** '!'**,** '@'**,** '#'**,** '$'**,** '%'**,** '^'**,** '&'**,** '\*'**,** '('**,** ')']  
 p=""  
 for x in range(len.get()):  
 p+=rm.choice(l1)  
 s.set(p)  
def clipboard():  
 rp=s.get()  
 pyperclip.copy(rp)  
Label(root**,**text="Password Creator"**,** font="arial 20 bold").pack()  
Label(root**,**text="Enter desired password length").pack(pady=**3**)  
Entry(root**,**textvariable=len).pack(pady=**3**)  
Button(root**,**text="Create Password"**,**command=pwd).pack(pady=**7**)  
Entry(root**,**textvariable=s).pack(pady=**3**)  
Button(root**,**text="Copy to clipboard"**,**command=clipboard).pack()  
Button(root**,**text="Exit program"**,**command=root.destroy).pack()  
root.mainloop()

***OUTPUT:***

***REFERENCES:***

* <https://en.wikipedia.org/wiki/Password_strength>
* Problem Solving and Programming with Python – Reema Thareja (for tkinter)
* COMPUTER SCIENCE with python (for class 12) – Sumita Arora (tkinter reference)
* <https://www.google.com/search?q=is+python+random+really+random&rlz=1C1GCEA_enIN902IN902&oq=is+python+random+really+random&aqs=chrome.0.69i59.1282j0j7&sourceid=chrome&ie=UTF-8>
* https://stackoverflow.com/questions/2145510/random-is-barely-random-at-all
* <https://docs.python.org/3/library/random.html>
* <https://interactivechaos.com/en/python/function/randomsystemrandom>