

# Course Name:

**Data Programming and Predictive Analysis for Business.**

# Project: (The Random Password Generator)

**Presented by: Marwa Medhat Ali Said.**

# Presented to: Ts. Dr. KHAW KHAI WAH

**Matrix code:(P-EM0388/22)**

# Email: [marwa.medhat@student.usm.my](mailto:marwa.medhat@student.usm.my)

**The Random Password Generator**

The aim of this report is to present random number generator code for generating a strong password for the user and the main objective of the report is to show:

* Algorithm development.
* Pseudocode.
* Flowcharts.
* Coding.
* Conclusion.

# Algorithm development:

* Greeting the user.
* Take number of characters from user.
* Validate the input between (8-16)
* If the length in range (8 -16):
  + Calculate each category length.
  + Pick a random sample of specified length from each category and pick a random. sample of the remaining length from all the categories.
  + Shuffle the resulting list.
  + Join the password list elements together to form the password string.
  + show the password.

# Pseudocode

Step1: Output (Hi). Step2: While True:

Step3: Ask the user for length:

Step4: If length not number or less than 8 or more than 16 then Show (The password must be integer and between (8-16))

Show (Try again)

If password length between (8-16) Break the loop

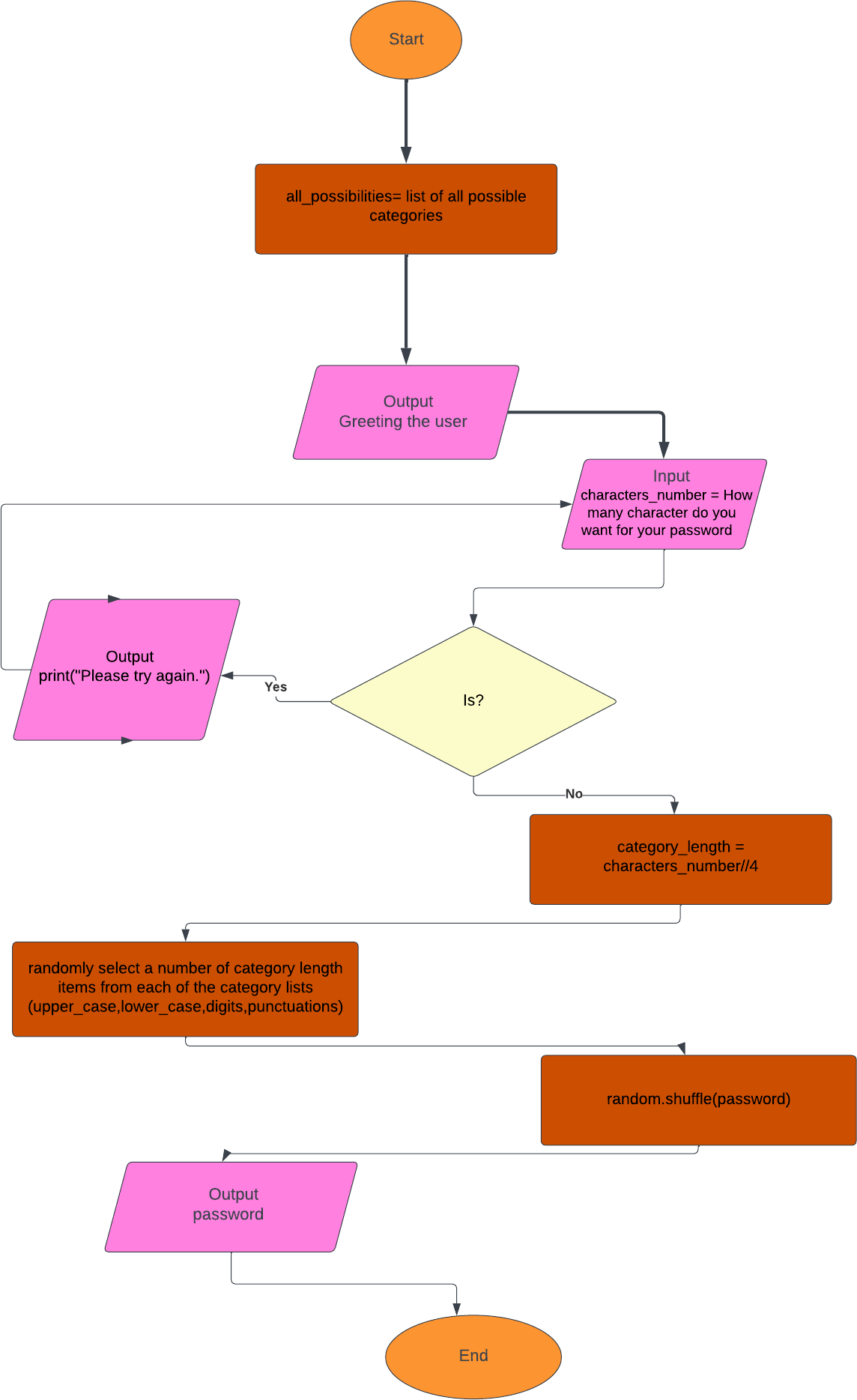
End while

Step5: calculate category \_length from this equation (characters \_number//4)

Step6: Create a password randomly (Picking a random digit from each category) Step7: Shuffle the password to randomize the character taken from each category. Step8: Convert password list to string.

Step9: Show the password for the user.

# flowchart



**5.Code**

import string import random

all\_possibilities= list(string.ascii\_lowercase + string.ascii\_uppercase + string.digits +string.punct uation)

# Greating the user print("Hi!")

while True: try:

characters\_number = int(input("How many character do you want for your password?! "))

if characters\_number < 8 or characters\_number>16 :

print("The password must have a minimum of 8 digits and maximum of 16 digits!") print("Please try again.")

else:

break except:

print("Please enter numbers only!")

#character\_number = input("How many characters do you want for your password?!") #inp ut added & variable name changed from characters\_number to character\_number#######

# Picking a random digits from each category category\_length = characters\_number//4

password = random.sample(string.ascii\_lowercase, category\_length) + random.sample(string.asc ii\_uppercase, category\_length) + random.sample(string.digits, category\_length) + random.sampl e(string.punctuation, category\_length) + random.sample(all\_possibilities, characters\_number(cat egory\_length\*4))

random.shuffle(password) password = "".join(password) print(password)

# python code(explanation)

this figure show running python code which take the length from user between (8,16) when the user input equal 9

# python code (Explanation)

* This figure show running python code which take the length from user between (8,16) when we asked the user for how many character do you want for your password?

The user enters: jdhfg

Error wrong input from the user (Please enter numbers only!)

* This figure show running python code which take the length from user between (8,16) when we asked the user for how many character do you want for your password?

The user enters: 177

Error wrong input from the user (The password must have a minimum of 8 digits and maximum of 16 digits!)

Please try again.

* This figure show running python code which take the length from user between (8,16) when we asked the user for how many character do you want for your password?

The user enters: 10

And the program suggest: n3Wt$6)G)



# 5.Conclusion

We use try and except to handle the mistakes from the user input and random number generator for various distributions are given in the Random module to select randomly.

When we want the user to have several times of trials, we can use (While) for looping asking the user for the input.

If statement (you want to check a condition if it is true to take a decision or not)

We use String module to help us to have characters (upper\_casse, lower\_case, digits, punctuations) from each group:

string.**ascii\_letters.**

# string.ascii\_letters are including ascii\_lowercase and ascii\_uppercase .

string.**ascii\_lowercase.**

The lowercase letters are: “abcdefghijklmnopqrstuvwxyz”. string. **ascii\_uppercase.**

The uppercase letters are: “ABCDEFGHIJKLMNOPQRSTUVWXYZ” string.**digits.**

The string digits are: “0123456789. string.**hexdigits**

The string.hexdigits: **. “0123456789abcdefABCDEF”.**

string.**punctuation**

string.**punctuation are** !#$%&'()\*+,-./:;<=>?@[\]^\_`{|}~ <https://docs.python.org/3/library/string.html>