PUSAT PENGAJIAN PENGURUSAN
SCHOOL OF MANAGEMENT

The Random password generator

Data Programming and Predictive Analytics for Business

Lecturer: Dr. Khaw Khai Wah

Submitted by: Alshaimaa Abdelghani

Metric number:P-EM0379/22

alshaimaa.abdelghany@student.usm.my

Semester 1
Academic Session 2021/2022
School of Management
Universiti Sains Malaysia

Table of Contents

Introduction:

Most of us, has been aware of hacking. Being hacked is so dangerous and is usually associated with bank accounts in the minds of regular personnel. A common scenario of being hack is losing the money by getting the bank account hacked. However, cybersecurity has been active lately to protect users from such a scenario.

Problem statement:

It is well known that strong passwords are a basic vital defense against being hacked. Passwords need to be saved and not easily guessed to be safe. It is also needed to be a strong password which means having different limitations or conditions as symbols and mixing different characters and numbers. To be able to achieve such a safe and strong password, users need a password generator that can generate random combination of characters respecting the limitations and within a needed length.

Study objectives:

1. Construct a program to generate a password, this program should cover the following consecutive objectives to be achieved.
2. Get the right length of password that is between the range 8 and 16 from the user.
3. Generate a random password that is hard to be guessed, to be safe.
4. Check the password is strong by making sure it has at least one symbol, one lower case and higher case letter and one number.
5. Print the strong and safe password to the user.

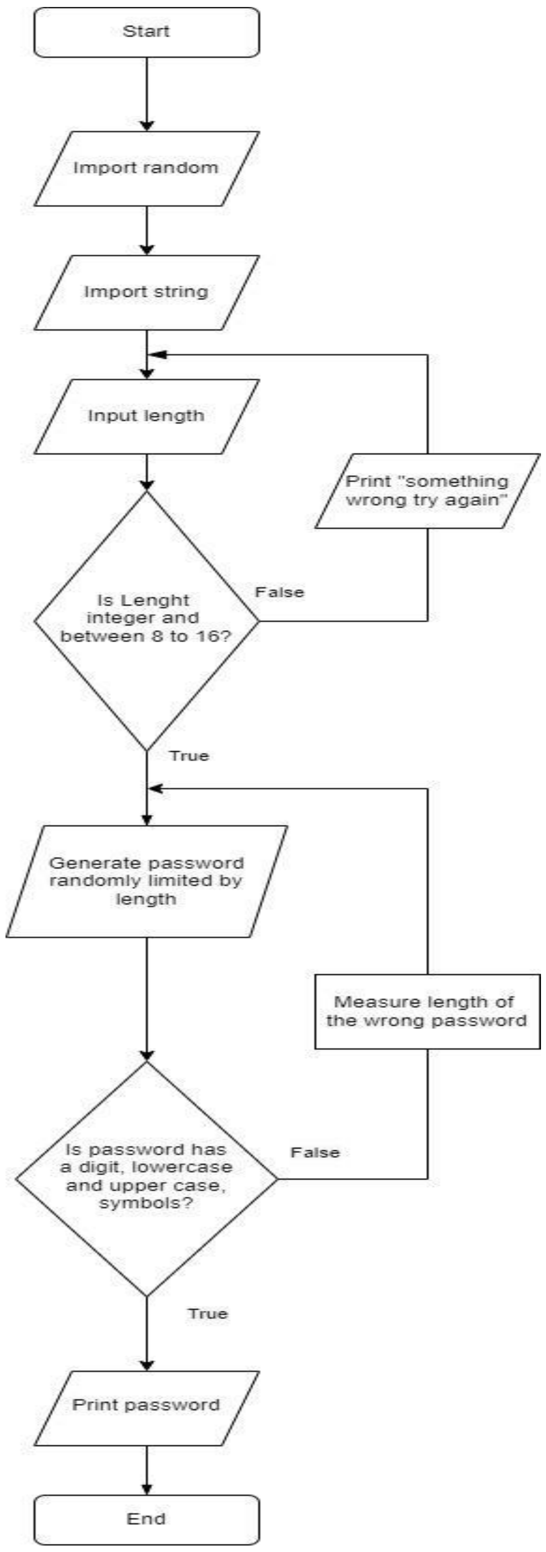
Algorithm development:

1. Greeting the user
2. Request Length of password
3. Check Length is integer and between 8 and 16
4. Declare variable seed that has a list of letters (lower and higher case), numbers and symbols.
5. Generate random password from the seed
6. Check the password contains at least one of the numbers, the symbols, upper-case and lower-case letters.
7. Print the password

Pseudocode Code:

```
Begin
Greet the user and Enter length
Check Length is an integer
If length is an integer
        If the length is in the range of 8 to 16
                Get a list of characters that contains numbers, letters (lower and upper case), symbols.
                Generate a list of randomly chosen characters limited by length
                Join the generated random list of characters for the password to be a single string
                Declare counter variable equals to zero
                Check password limitations which are at least one of the numbers, the symbols, upper-case and lower-case letters
                Add to the counter if it meets each limitation
                If the counter equals Four
                Print the password
                Else Re-generate the password
                Endif
        Else Re-enter the length
        print ("Something wrong, check password length is between 8 and 16!")
        Endif
Else Re-enter the length
Print "something wrong and enter it again", If length not within range
Endif
```

Flowchart:


Start
Import random
Import string
Input length
Print "something wrong try again"
Is Lenght integer and between 8 to 16?
False
True
Generate password randomly limited by length
Measure length of the wrong password
Is password has a digit, lowercase and upper case, symbols?
False
True
Print password
End

Python Code:

```
import string
import random

def generate_password(L):
  Numbers = string.digits
  Letters = string.ascii_letters
  Symbols = string.punctuation
  seed = Numbers+Letters+Symbols
  password = random.sample(seed, k=L)
  password = "".join(password)
  counter = 0
  if True in [i.isdigit() for i in password]:
    counter += 1
  if True in [i.islower() for i in password]:
    counter += 1
  if True in [i.isupper() for i in password]:
    counter += 1
  if True in [i in Symbols for i in password]:
    counter += 1
  if counter == 4:
    print(password)
  else:
    generate_password(len(password))

def main():
  while True:
    length = input("Hello, Would you tell us the password length? (Between 8-16) ")
    try:
      length = int(length)
      if length in range(8, 17):
        password = generate_password(length)
        break
      else:
        print("Something wrong, check password length is between 8 and 16!")
    except ValueError:
      print("Something wrong, Program accept numbers only!")

main()
```

Conclusion:

I have applied most of what i have learnt during the lectures

1. Algorithms help make a blueprint to the program, main steps and how to solve the problem into code, and I found out my favorite one is pseudocode, it help me keep track of the code and be organized to know what the next step should be.

2. The harder the problem and the implementation, the higher the probability to learn new code syntax and to go out of the box!. I could combine different techniques and code portions to be able to solve the problem of checking the limitations

3. I am more appreciative towards modules and how much effort and time has been saved. As I have learnt how to read the modules, understand syntax, try out the ones I suspect would help me, then become really familiar with it!

4. Looping help the code keep going without being interrupted.

5. Try and except is very helpful for handling the errors.