UNIVERSIDAD
DE SANTIAGO
DE CHILE

Guía 10.2

10145 — Fundamentos de Programación para Ingenería
10110 — Fundamentos de Computación y Programación

Universidad de Santiago de Chile

Primer Semestre 2024

Instrucciones Generales

- Cree un `.py` con su `RUT` como nombre del archivo.
- Agregue como encabezado del programa los siguientes datos:

```
# FUNDAMENTOS DE PROGRAMACIÓN PARA INGENIERÍA/
# FUNDAMENTOS DE COMPUTACIÓN Y PROGRAMACIÓN
# SECCIÓN DEL CURSO:
# PROFESOR DE TEORÍA:
# PROFESOR DE LABORATORIO:
#
# AUTOR
# NOMBRE:
# RUN:
# CARRERA:
```

Ejercicio 10.2
Generador de Contraseñas

Utilice el revisador — estudiante 10.2
La seguridad informática es muy importante, pues nuestros datos son los que están expuestos, si no somos cuidadosos y los protegemos detrás de contraseñas seguras.
Una forma de generar contraseñas es a través de procesos aleatorios o pseudoaleatorios. Utilizando el módulo `random`, genere una contraseña aleatoria a partir de dos parámetros, que indican el largo máximo y el largo mínimo de esta (en ese orden), donde el mínimo es 8 por defecto. La función debe llevar por nombre `gen_pw` y cumplir con ciertas reglas para su resultado.

Ejercicio 10.2
Reglas de la contraseña

- Debe tener al menos una letra minúscula de la tabla ASCII (es decir, sin tildes).
- Debe tener al menos una letra mayúscula de la tabla ASCII (es decir, sin tildes).
- Debe tener al menos un número.
- Debe tener al menos un símbolo entre `?`, `!`, `@`, `#`, `$`, `%`, `^`, `&` y `*`.
- La cantidad de caracteres debe estar entre los parámetros entregados.
- No puede tener la misma cantidad de tipos de caracteres (es decir, si la clave tiene ocho caracteres, no puede tener dos mayúsculas, dos minúsculas, dos números y dos símbolos).

Ejercicio 10.2
Entrada y Salida

Entrada

La función debe tener dos entradas, una obligatoria, correspondiente a un número entero que representa la cantidad máxima de caracteres en la contraseña a generar, y otro opcional, con la cantidad mínima, que por defecto debe ser 8.

Salida

Un *string* con la contraseña generada, cumpliendo las reglas señaladas previamente.

Ejercicio 10.2
Consideraciones

- Si bien la mejor alternativa para generar secuencias criptográficamente seguras en Python es la biblioteca `secrets`, en este ejercicio se utilizará la biblioteca `random`.
- La evaluación de las contraseñas generadas en este problema, como son aleatorias, será con mecanismos que determinen la validez de la respuesta entregada, en lugar de comparación de respuestas exactas.
- Si la cantidad máxima de caracteres es menor que la cantidad mínima, se debe entregar un *string* vacío como respuesta.

Ejercicio 10.2
Ejemplos

Entrada:

```
for i in range(5):
    print(gen_pw(10))
```

Salida:

```
KzVKd8O^
VDs!0ew$r&
?6?P0zF6@
7xDf3@17
w?&J402p!
```

Entrada:

```
for i in range(5):
    print(gen_pw(16, 10))
```

Salida:

```
%s%VJ10*sG^
3zi*CS&7$?h^
g%9*!!&B3P
E&RW9v1509UCQ1t
!!?LJbWX44Q?3n
```