

IWI-131 Programación: Tarea UVAs 4 y 5 - Ciclos y Strings



Una contraseña segura es la primera línea de defensa contra el acceso no autorizado a tus datos personales y a tus cuentas de servicios. A medida que los métodos de ataque se vuelven más sofisticados, es crucial fortalecer nuestras contraseñas para proteger nuestra información. En esta tarea usted debe escribir un programa para evaluar el nivel de seguridad de las contraseñas que propone un usuario.

En las contraseñas no se permiten espacios en blanco. Sólo pueden contener letras mayúsculas y minúsculas (incluyendo las vocales con tilde, la ñ y la Ñ), dígitos numéricos y estos caracteres especiales: `¡ ! ¿ ? @ # $ % & _ - + * = | { } [] . , ; : < > / \ ~ ^ ' "`

Una contraseña se considera segura si:

- Tiene 10 o más caracteres.
- Contiene una o más letras minúsculas.
- Contiene una o más letras mayúsculas.
- Contiene uno o más dígitos numéricos.
- Contiene uno o más caracteres especiales.
- No contiene el nombre del usuario ni el de su mascota.
- No contiene patrones de repetición de 5 letras o 5 dígitos iguales, consecutivos.

El programa debe comportarse de la forma indicada en los ejemplos. Al inicio solicita el nombre de la persona y de su mascota, una única vez. Posteriormente pide propuestas de contraseña y las evalúa, indicando criterios que no se cumplen, cuando corresponda. No es necesario indicar todos los criterios no cumplidos, basta con uno. En caso de cumplir todos los criterios, debe indicar que la contraseña es válida. El proceso termina cuando se ingresa el texto `FIN` en vez de una contraseña.

No hay un orden particular en que se deban probar los criterios de seguridad. Lo importante es que no se acepten contraseñas que no sean seguras.

Recuerde que el trabajo debe ser estrictamente individual.

Restricciones

- Puede utilizar **únicamente** la materia estudiada en la asignatura, hasta la UVA 5.
- No está permitido utilizar listas, tuplas o diccionarios.
- No está permitido el uso de `index`, `find`, `split`, `join` y `format` sobre *strings*.
- Se debe utilizar ciclos para procesar los *strings*.

Ejemplo de ejecución (Las entradas del usuario se destacan en negrita)

```
Nombre: Bob
Nombre de su mascota: Gary
Contraseña propuesta (FIN para terminar): password
Contraseña muy corta, debe tener al menos 10 caracteres.
Contraseña propuesta (FIN para terminar): mipassword
Debe incluir letras mayúsculas.
Contraseña propuesta (FIN para terminar): Mipassword
Debe incluir dígitos numéricos.
Contraseña propuesta (FIN para terminar): Mipassword2
Debe incluir caracteres especiales válidos.
Contraseña propuesta (FIN para terminar): MipasswordÇ2
La contraseña contiene caracteres inválidos.
Contraseña propuesta (FIN para terminar): Mipassword.2
Contraseña válida!
Contraseña propuesta (FIN para terminar): Password_boby_2
Contraseña no debe incluir su nombre.
Contraseña propuesta (FIN para terminar): password.GARY.2024
Contraseña no debe incluir el nombre de su mascota.
Contraseña propuesta (FIN para terminar): password.GA-RY.2024
Contraseña válida!
Contraseña propuesta (FIN para terminar): Hola55555!mundo
Contraseña no debe contener patrones: 55555
Contraseña propuesta (FIN para terminar): Hola5!mundooooo
Contraseña no debe contener patrones: ooooo
Contraseña propuesta (FIN para terminar): Hola5!mundooooo
Contraseña válida!
Contraseña propuesta (FIN para terminar): FIN
```