Task 6 –

Creating multiple password with varying complexity

ax7yBPCH~hz4ir8F

L*<jf2:#lU5u\6JH

El%NAh8<diP?g!4z

VIC^4>OR<0/_#9sy

pa}SQ,4/5G<k?o).

0UTX$Lv2YKsFgzSV

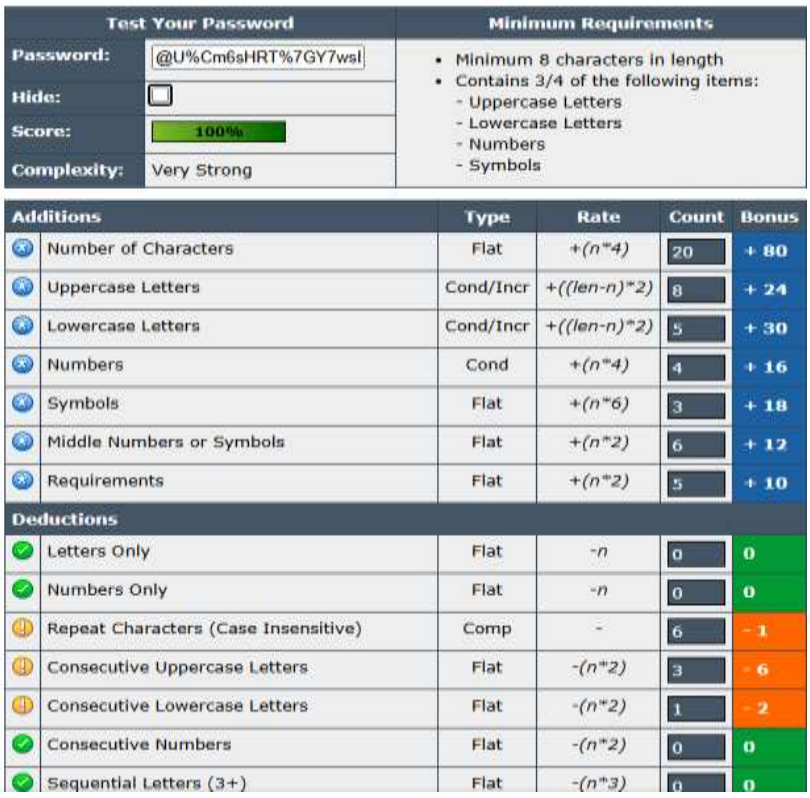
5FlE?QAqfOtRT@Hk

5X?hL2sZ3t1$U@Ta

7q,x+txSYNB=tqXTM9

@U%Cm6sHRT%7GY7wsB1h

Testing password strength



Test Your Password		Minimum Requirements
Password:	@U%Cm6sHRT%7GY7wsl	• Minimum 8 characters in length • Contains 3/4 of the following items: - Uppercase Letters - Lowercase Letters - Numbers - Symbols
Hide:	☐	
Score:	100%	
Complexity:	Very Strong	

Additions	Type	Rate	Count	Bonus
Number of Characters	Flat	+(n*4)	20	+ 80
Uppercase Letters	Cond/Incr	+((len-n)*2)	8	+ 24
Lowercase Letters	Cond/Incr	+((len-n)*2)	5	+ 30
Numbers	Cond	+(n*4)	4	+ 16
Symbols	Flat	+(n*6)	3	+ 18
Middle Numbers or Symbols	Flat	+(n*2)	6	+ 12
Requirements	Flat	+(n*2)	5	+ 10
Deductions				
Letters Only	Flat	-n	0	0
Numbers Only	Flat	-n	0	0
Repeat Characters (Case Insensitive)	Comp	-	6	- 1
Consecutive Uppercase Letters	Flat	-(n*2)	3	- 6
Consecutive Lowercase Letters	Flat	-(n*2)	1	- 2
Consecutive Numbers	Flat	-(n*2)	0	0
Sequential Letters (3+)	Flat	-(n*3)	0	0

- $tr0ngP@55w0rd!
- Br!ght&SunnyD@ys24
- MysT3r!ous#C@stle9
- P@55w0rd#SecUr3!!
- Tr@v3lL0ver$#2022
- 7H@ppy#D@ys!2023
- QwErTy&123$!@
- Bl@ckH0l3$&G@l@xy
- ^Dr@g0n$&M@g1c^!
- SuPer!S@f3#P@ss
- *UnBr3@k@bl3*12
- P!an0M@st3r#78
- B3yond$th3$e@!!
- S@f3H@ven#999
- J@va&Pyth0nR0ck!

Common Mistakes in Password Creation

1. **Using Simple and Predictable Passwords:**
    - Avoid easily guessable passwords like “123456,” “password,” or “qwerty.”
2. **Reusing Passwords:**
    - Using the same password across multiple accounts increases vulnerability. If one account is compromised, all accounts using the same password are at risk.
3. **Using Personal Information:**
    - Avoid passwords that include names, birthdays, or easily accessible personal information.
4. **Short Passwords:**
    - Passwords that are too short are easier to crack. Aim for at least 12 characters when possible.

5. **Lack of Character Variety:**
    - Failing to include a mix of uppercase and lowercase letters, numbers, and special characters reduces password strength.
6. **Using Common Words or Phrases:**
    - Avoid using common dictionary words or popular phrases, even with character substitutions.
7. **Patterns and Sequences:**
    - Avoid using predictable patterns like “abcd1234” or “password1.”
8. **Not Changing Passwords Regularly:**
    - Regularly updating passwords helps protect against unauthorized access, especially if an old password has been compromised.
9. **Ignoring Two-Factor Authentication (2FA):**
    - Failing to enable 2FA adds an extra layer of security beyond just the password.
10. **Writing Down Passwords:**
    - Avoid writing passwords on paper or storing them in plain text files. Use a password manager instead.

Creating a Strong 8-Character Password

1. **Mix Character Types:**
    - Include at least one uppercase letter, one lowercase letter, one number, and one special character.
    - Example: **A3d#8kLm**
2. **Avoid Common Words:**
    - Do not use easily guessable words, names, or simple sequences.
    - Avoid using personal information like birthdays or names.
3. **Use Randomness:**
    - Combine characters in an unpredictable manner.
    - Example: **Xy7!Za9Q**
4. **Substitute Letters and Numbers:**

- Replace common letters with similar-looking numbers or symbols.
- Example: Replace ‘S’ with ‘5’ or ‘A’ with ‘@’.

5. **Avoid Repeated Patterns:**
   - Do not use repeating characters or sequences like ‘1234’ or ‘aaaa’.
   - Example: Avoid using patterns such as **P@ssP@ss**.
6. **Utilize a Password Manager:**
   - Use a password manager to generate and store complex passwords.
   - This helps create truly random and strong passwords without memorizing them.

Test Your Password		Minimum Requirements
Password:	ax7yBPCH~hz4ir8F	• Minimum 8 characters in length • Contains 3/4 of the following items: - Uppercase Letters - Lowercase Letters - Numbers - Symbols
Hide:	☐	
Score:	100%	
Complexity:	Very Strong	

Additions	Type	Rate	Count	Bonus
Number of Characters	Flat	+(n*4)	16	+ 64
Uppercase Letters	Cond/Incr	+((len-n)*2)	5	+ 22
Lowercase Letters	Cond/Incr	+((len-n)*2)	7	+ 18
Numbers	Cond	+(n*4)	3	+ 12
Symbols	Flat	+(n*6)	1	+ 6
Middle Numbers or Symbols	Flat	+(n*2)	4	+ 8
Requirements	Flat	+(n*2)	5	+ 10
Deductions				
Letters Only	Flat	-n	0	0
Numbers Only	Flat	-n	0	0
Repeat Characters (Case Insensitive)	Comp	-	0	0
Consecutive Uppercase Letters	Flat	-(n*2)	3	- 6
Consecutive Lowercase Letters	Flat	-(n*2)	3	- 6
Consecutive Numbers	Flat	-(n*2)	0	0
Sequential Letters (3+)	Flat	-(n*3)	0	0

Identify best practices for creating strong passwords

Creating strong passwords is a cornerstone of online security. Here are the best practices:

1. Length is Key:

- Aim for at least 12 characters, but 14 or more is even better. Longer passwords are significantly harder to crack through brute-force attacks.

2. Mix it Up (Complexity):

- **Combine different character types:** Use a blend of uppercase letters, lowercase letters, numbers, and special characters (e.g., !, @, #, $, %, ^, &, *).
- **Avoid predictable patterns:** Don't use sequential numbers (1234), keyboard patterns (qwerty), or simple repetitions (aaaa).

3. Uniqueness is Paramount:

- **Use a different password for every account:** If one account is compromised, attackers won't be able to access your other accounts. This is perhaps the most crucial practice.
- **Never reuse passwords:** Especially for important accounts like email and banking.

4. Avoid Personal Information and Common Words:

- **Don't use easily guessable info:** This includes your name, nickname, initials, birth date, phone number, pet's name, street name, or any information easily found on your social media profiles.

- **Steer clear of dictionary words and common phrases:** Hackers use "dictionary attacks" that try common words and phrases. Even substitutions like "P@ssword" for "password" are often easily guessed.
- **Don't use song lyrics, movie titles, or quotes:** These are often too common and can be easily found.

When discussing password security, "dictionary attacks" and "brute-force attacks" are two common, yet distinct, methods cybercriminals use to try and gain unauthorized access to accounts. Here's a breakdown:

Dictionary Attack

What it is: A dictionary attack is a type of brute-force attack that attempts to crack a password by systematically trying every word in a pre-defined list (often called a "wordlist" or "dictionary") as a potential password. This list typically includes common words, phrases, names, popular password variations (e.g., "password123", "qwerty"), and even passwords exposed in previous data breaches.

How it works:

1. **Compiler a Wordlist:** Attackers create or acquire large lists of potential passwords. These lists can be generated from actual dictionaries, common literary works, leaked password databases, or even public information related to a target (like company names, sports teams, or common pet names).
2. **Automated Guessing:** Specialized software automates the process of entering these words as passwords, often trying various permutations (e.g., capitalizing the first letter, adding numbers or symbols).
3. **Exploiting Human Tendencies:** This method is effective because many people choose simple, memorable passwords that are often found in dictionaries or common phrases.

Key Characteristics:

- **Targeted:** Focuses on a subset of likely passwords.
- **Faster:** Generally quicker than a full brute-force attack because it's not trying every single combination.
- **Relies on Predictability:** Success hinges on users choosing predictable passwords.

Brute-Force Attack

What it is: A brute-force attack is a trial-and-error method used to crack passwords, encryption keys, or other credentials by systematically trying *every possible combination* of

characters until the correct one is found. It's an exhaustive search that doesn't rely on intelligence or lists, but rather on sheer computational power.

How it works:

1. **Define Character Set:** The attacker defines the character set to be used (e.g., lowercase letters, uppercase letters, numbers, special characters).
2. **Generate Combinations:** Automated software (often using powerful computing resources like GPUs or botnets) generates and tests every possible combination of characters within the defined set, from the shortest possible length up to a specified maximum.
3. **Time-Consuming (for strong passwords):** The time it takes to succeed depends exponentially on the length and complexity of the password. A short, simple password can be cracked in minutes or seconds, while a long, complex one could take years, decades, or even millennia with current technology.

Key Characteristics:

- **Exhaustive:** Attempts every single possible combination.
- **Resource-Intensive:** Requires significant computational power and time, especially for longer passwords.
- **Guaranteed (eventually):** Given enough time and resources, a brute-force attack will eventually crack any password, regardless of its strength (though "eventually" can mean an astronomically long time).
- **Broader Category:** Dictionary attacks are often considered a *type* of brute-force attack, albeit a more refined one.

Key Differences and Relationship

- **Scope of Guesses:**
    - **Dictionary Attack:** Tries a *pre-compiled list* of common or known words and patterns.
    - **Brute-Force Attack:** Tries *every possible combination* of characters.
- **Efficiency:**
    - **Dictionary Attack:** More efficient and targeted, but only works if the password is on the list or a variation of it.
    - **Brute-Force Attack:** Less efficient due to its exhaustive nature, but theoretically guaranteed to find the password given enough time.

- **Relationship:** A dictionary attack is a *subset* or *specialized form* of a brute-force attack. It makes the brute-force process more efficient by focusing on the most probable guesses first. Hybrid attacks combine dictionary words with brute-force elements (e.g., adding numbers or symbols to dictionary words).

summarize how password complexity affects security

Password complexity directly affects security by influencing how difficult it is for an attacker to guess or crack a password. Here's a summary of its impact:

- **Increased Search Space for Brute-Force Attacks:** A more complex password, which includes a wider range of character types (uppercase, lowercase, numbers, special characters), dramatically increases the number of possible combinations an attacker would need to try in a brute-force attack. For example, an 8-character password with only lowercase letters has far fewer possibilities than an 8-character password using all four character types. Every added character type and every additional character significantly multiplies the effort required for an attacker.
- **Resistance to Dictionary Attacks:** Complexity rules often discourage the use of common words, names, or easily guessable patterns. By requiring a mix of character types and avoiding personal information or dictionary words, complex passwords become resistant to dictionary attacks, which rely on pre-compiled lists of common passwords.
- **Higher Entropy:** In information theory, entropy measures the randomness or unpredictability of a password. Complex passwords, by incorporating diverse character sets and avoiding predictable patterns, have higher entropy, meaning they are much harder to guess randomly.
- **Deterrence of Attackers:** The sheer time and computational resources required to crack a truly complex password can deter attackers. They may move on to easier targets if a password appears too difficult to compromise.

However, there's a crucial **usability trade-off** that security experts, including NIST (National Institute of Standards and Technology), have increasingly emphasized:

- **User Frustration and Poor Password Hygiene:** Overly strict or convoluted complexity requirements can lead to user frustration. This often results in users adopting insecure practices, such as:
    - **Predictable Alterations:** Taking a simple word and adding predictable numbers or symbols (e.g., "password" becoming "P@ssword1!").
    - **Writing Passwords Down:** Storing complex, hard-to-remember passwords in insecure locations.

- **Password Reuse:** Using the same complex password across multiple accounts, which undermines security if one account is compromised.
- **Choosing Easily Guessable Passwords (Despite Rules):** Users may still choose passwords that meet the complexity rules but are based on easily discoverable personal information.