Brute Force Attack

CODE:

```
import itertools
import string


def bruteforce_attack(password):
    chars = string.printable.strip()
    attempts = 0
    for length in range(1, len(password) + 1):
        for guess in itertools.product(chars, repeat=length):
            attempts += 1
            guess = ''.join(guess)
            if guess == password:
                return (attempts, guess)
    return (attempts, None)


password = input("Input the password to crack: ")
attempts, guess = bruteforce_attack(password)
if guess:
    print(f"Password cracked in {attempts} attempts. The password is {guess}.")
else:
    print(f"Password not cracked after {attempts} attempts.")
```

OUTPUT:

Input the password to crack: pass

Password cracked in 21695135 attempts. The password is pass.

Input the password to crack: ade#

Password cracked in 9261603 attempts. The password is ade#.

Dictionary Attack

CODE:

```
import hashlib

# List of commonly used passwords and their variations

common_passwords = ["password", "password123", "letmein", "qwerty", "123456", "abc123", "admin", "welcome", "monkey", "sunshine"]

password_variations = ["", "123", "1234", "12345", "123456", "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "-", "_", "+", "=", "/", "\\", "|", "[", "]", "{", "}", "<", ">"]

# Hash of the password to be attacked

hashed_password = hashlib.sha256(b"mypass12#@").hexdigest()

# Try out all possible combinations of common passwords and their variations

for password in common_passwords:
    for variation in password_variations:
        possible_password = password + variation
        hashed_possible_password = hashlib.sha256(possible_password.encode()).hexdigest()
        if hashed_possible_password == hashed_password:
            print(f"Password found: {possible_password}")
            break
    else:
        continue
```

```
    break
else:
    print("Password not found")
```

OUTPUT:

Password not found