#!/usr/bin/env python3

"""

Password Generator (secure, CLI)

Usage examples:

python password\_gen.py # one 16-char strong password

python password\_gen.py -l 24 -n 5 # 5 passwords, length 24

python password\_gen.py --no-symbols # only letters+digits

python password\_gen.py -l 20 --copy # copy first result to clipboard (needs pyperclip)

"""

import argparse

import secrets

import string

import sys

AMBIGUOUS = set("Il1O0|`'\";:,.()[]{}")

DEFAULT\_SYMBOLS = "!@#$%^&\*\_-+=?/~"

def build\_pool(use\_lower, use\_upper, use\_digits, use\_symbols, exclude\_ambiguous):

pools = []

if use\_lower:

pools.append(set(string.ascii\_lowercase))

if use\_upper:

pools.append(set(string.ascii\_uppercase))

if use\_digits:

pools.append(set(string.digits))

if use\_symbols:

pools.append(set(DEFAULT\_SYMBOLS))

if not pools:

raise ValueError("At least one character category must be enabled.")

if exclude\_ambiguous:

pools = [p - AMBIGUOUS for p in pools]

# Flatten to a single allowed set

allowed = set().union(\*pools)

if not allowed:

raise ValueError("Character pool is empty after exclusions.")

return pools, list(allowed)

def generate\_password(length, pools, allowed):

if length < len(pools):

raise ValueError(f"Length must be at least {len(pools)} to include all categories.")

# Ensure at least one char from each selected pool

pw\_chars = [secrets.choice(list(p)) for p in pools]

# Fill the rest from the full allowed set

remaining = length - len(pools)

pw\_chars += [secrets.choice(allowed) for \_ in range(remaining)]

# Shuffle securely

for i in range(len(pw\_chars) - 1, 0, -1):

j = secrets.randbelow(i + 1)

pw\_chars[i], pw\_chars[j] = pw\_chars[j], pw\_chars[i]

return "".join(pw\_chars)

def main():

parser = argparse.ArgumentParser(description="Generate secure random passwords.")

parser.add\_argument("-l", "--length", type=int, default=16, help="Password length (default: 16)")

parser.add\_argument("-n", "--number", type=int, default=1, help="How many passwords to generate (default: 1)")

group = parser.add\_argument\_group("Character choices")

group.add\_argument("--no-lower", action="store\_true", help="Exclude lowercase letters")

group.add\_argument("--no-upper", action="store\_true", help="Exclude uppercase letters")

group.add\_argument("--no-digits", action="store\_true", help="Exclude digits")

group.add\_argument("--no-symbols", action="store\_true", help="Exclude symbols")

parser.add\_argument("-a", "--no-ambiguous", action="store\_true",

help="Exclude ambiguous characters (e.g., 0/O, 1/l/I)")

parser.add\_argument("--copy", action="store\_true",

help="Copy the first generated password to the clipboard (requires pyperclip)")

args = parser.parse\_args()

try:

pools, allowed = build\_pool(

use\_lower=not args.no\_lower,

use\_upper=not args.no\_upper,

use\_digits=not args.no\_digits,

use\_symbols=not args.no\_symbols,

exclude\_ambiguous=args.no\_ambiguous

)

passwords = [generate\_password(args.length, pools, allowed) for \_ in range(args.number)]

except ValueError as e:

print(f"Error: {e}", file=sys.stderr)

sys.exit(1)

for pw in passwords:

print(pw)

if args.copy and passwords:

try:

import pyperclip # pip install pyperclip

pyperclip.copy(passwords[0])

# Don’t print the password again; just confirm copy

print("(First password copied to clipboard.)", file=sys.stderr)

except Exception:

print("(Clipboard copy failed. Install with: pip install pyperclip)", file=sys.stderr)

if \_\_name\_\_ == "\_\_main\_\_":

main()