import java.security.SecureRandom;

import java.util.Scanner;

public class PasswordGenerator {

private static final String UPPERCASE = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

private static final String LOWERCASE = "abcdefghijklmnopqrstuvwxyz";

private static final String NUMBERS = "0123456789";

private static final String SPECIAL\_CHARACTERS = "!@#$%^&\*()-\_=+<>?";

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the desired password length: ");

int length = scanner.nextInt();

System.out.print("Include uppercase letters? (y/n): ");

boolean includeUppercase = scanner.next().toLowerCase().charAt(0) == 'y';

System.out.print("Include lowercase letters? (y/n): ");

boolean includeLowercase = scanner.next().toLowerCase().charAt(0) == 'y';

System.out.print("Include numbers? (y/n): ");

boolean includeNumbers = scanner.next().toLowerCase().charAt(0) == 'y';

System.out.print("Include special characters? (y/n): ");

boolean includeSpecialChars = scanner.next().toLowerCase().charAt(0) == 'y';

String password = generatePassword(length, includeUppercase, includeLowercase, includeNumbers, includeSpecialChars);

System.out.println("Generated Password: " + password);

}

public static String generatePassword(int length, boolean includeUppercase, boolean includeLowercase, boolean includeNumbers, boolean includeSpecialChars) {

StringBuilder characterPool = new StringBuilder();

if (includeUppercase) {

characterPool.append(UPPERCASE);

}

if (includeLowercase) {

characterPool.append(LOWERCASE);

}

if (includeNumbers) {

characterPool.append(NUMBERS);

}

if (includeSpecialChars) {

characterPool.append(SPECIAL\_CHARACTERS);

}

if (characterPool.length() == 0) {

throw new IllegalArgumentException("At least one character set must be selected");

}

SecureRandom random = new SecureRandom();

StringBuilder password = new StringBuilder();

for (int i = 0; i < length; i++) {

int index = random.nextInt(characterPool.length());

password.append(characterPool.charAt(index));

}

return password.toString();

}

}