

```

# =====
# پروژه 7 :تولیدکننده و رمزگشایی رمز عبور
# =====

#وارد کردن کتابخانه های مورد نیاز
# random: برای انتخاب تصادفی کاراکترها
# string: برای دسترسی به مجموعه کاراکترهای آماده )،حروف اعداد(
import random
import string

#=====تابع تولید رمز عبور تصادفی=====
def generate_password(length=8, use_upper=True, use_lower=True, use_digits=True, use_symbols=False):
    """
    این تابع یک رمز عبور تصادفی تولید می‌کند.
    پارامترها:
    - length: طول رمز عبور (پیشفرض 8)
    - use_upper: True (پیشفرض) آیا حروف بزرگ استفاده شود؟
    - use_lower: True (پیشفرض) آیا حروف کوچک استفاده شود؟
    - use_digits: True (پیشفرض) آیا اعداد استفاده شود؟
    - use_symbols: False (پیشفرض) آیا نمادها استفاده شود؟

    رشته خالی برای جمع‌آوری کاراکترهای مجاز
    characters = ""

    #اضافه کردن حروف بزرگ (A-Z) (اگر کاربر خواسته باشد)
    if use_upper:
        characters += string.ascii_uppercase

    #اضافه کردن حروف کوچک (a-z) (اگر کاربر خواسته باشد)
    if use_lower:
        characters += string.ascii_lowercase

    #اضافه کردن اعداد (0-9) (اگر کاربر خواسته باشد)
    if use_digits:
        characters += string.digits

    #اضافه کردن نمادها اگر کاربر خواسته باشد
    if use_symbols:
        characters += "!@#$%^&*"

    #اگر هیچ نوع کاراکتری انتخاب نشده باشد
    if characters == "":
        return "خطا: حداقل یک نوع کاراکتر انتخاب کنید"

    #تولید رمز عبور
    password = ""

```

```

for i in range(length):
    # random.choice: انتخاب می‌کند یک کاراکتر تصادفی از رشته
    password += random.choice(characters)

return password

=====تابع ارزیابی قدرت رمز عبور=====
def check_strength(password):
    """
    این تابع قدرت رمز عبور را ارزیابی می‌کند.
    معیارها ( هر کدام 1 امتیاز):
    - طول >= 8
    - طول >= 12
    - وجود حروف بزرگ
    - وجود حروف کوچک
    - وجود اعداد
    - وجود نمادها

    """

    score = 0 # امتیاز اولیه صفر

    #متغیرهای بولین برای بررسی وجود انواع کاراکترها
    has_upper = False # آیا حرف بزرگ دارد؟
    has_lower = False # آیا حرف کوچک دارد؟
    has_digit = False # آیا عدد دارد؟
    has_symbol = False # آیا نماد دارد؟

    #بررسی تک تک کاراکترهای رمز عبور
    for ch in password:
        if ch.isupper(): # isupper(): بررسی حرف بزرگ بودن
            has_upper = True
        elif ch.islower(): # islower(): بررسی حرف کوچک بودن
            has_lower = True
        elif ch.isdigit(): # isdigit(): بررسی عدد بودن
            has_digit = True
        else: # اگر هیچکدام ،نبود یعنی نماد است
            has_symbol = True

    #امتیازدهی بر اساس طول
    if len(password) >= 8:
        score += 1
    if len(password) >= 12:
        score += 1

    #امتیازدهی بر اساس نوع کاراکترها
    if has_upper:
        score += 1

```

```

if has_lower:
    score += 1
if has_digit:
    score += 1
if has_symbol:
    score += 1

#تعیین سطح قدرت بر اساس امتیاز
if score >= 5:
    return "بسیار قوی"
elif score >= 4:
    return "قوی"
elif score >= 3:
    return "متوسط"
elif score >= 2:
    return "ضعیف"
else:
    return "بسیار ضعیف"

=====تابع رمزگذاری ساده=====
def simple_encode(text, shift=3):
    """
    این تابع یک متن ساده را رمزگذاری می‌کند.
    روش: هر کاراکتر را با کد عددی + shift تغییر می‌دهد.
    """
    result = ""
    for ch in text:
        # ord(ch): کد عددی کاراکتر را برمی‌گرداند (مثل ASCII)
        # chr(): عدد را دوباره به کاراکتر تبدیل می‌کند
        result += chr(ord(ch) + shift)
    return result

=====تابع رمزگشایی=====
def simple_decode(text, shift=3):
    """
    این تابع یک متن رمزگذاری شده را رمزگشایی می‌کند.
    روش: هر کاراکتر را با کد عددی - shift برمی‌گرداند.
    """
    result = ""
    for ch in text:
        result += chr(ord(ch) - shift)
    return result

=====تابع نمایش منوی اصلی=====
def show_menu():
    print("\n" + "=" * 45)
    print("تولیدکننده و رمزگشایی رمز عبور")

```

```

print("=" * 45)
print("1. تولید رمز عبور تصادفی")
print("2. ارزیابی قدرت رمز عبور")
print("3. رمزگذاری متن (Encode)")
print("4. رمزگشایی متن (Decode)")
print("5. خروج")

=====حلقه اصلی برنامه=====

while True:
    show_menu()

    try:
        choice = int(input("انتخاب شما ( 1-5 ): "))

        if choice == 5:
            print("...خروج از برنامه")
            break

        if choice == 1:
            print("\n--- تولید رمز عبور ---")
            length = int(input("طول رمز عبور (پیش فرض 8): ") or 8)
            use_upper = input("حروف بزرگ داشته باشد؟ (y/n): ").lower() == 'y'
            use_lower = input("حروف کوچک داشته باشد؟ (y/n): ").lower() == 'y'
            use_digits = input("اعداد داشته باشد؟ (y/n): ").lower() == 'y'
            use_symbols = input("نمادها داشته باشد؟ (y/n): ").lower() == 'y'

            password = generate_password(length, use_upper, use_lower, use_digits,
            print(f"\nرمز عبور تولید شده: {password}")
            print(f"قدرت رمز: {check_strength(password)}")

            elif choice == 2:
                password = input("\nرمز عبور مورد نظر را وارد کنید: ")
                strength = check_strength(password)
                print(f"قدرت رمز عبور: {strength}")

                if strength == "ضعیف" or strength == "بسیار ضعیف":
                    print("از ترکیب حروف، بزرگ، کوچک اعداد و نمادها استفاده کنید")

            elif choice == 3:
                text = input("\nمتن مورد نظر برای رمزگذاری: ")
                encoded = simple_encode(text)
                print(f"متن رمزگذاری شده: {encoded}")

            elif choice == 4:
                text = input("\nمتن رمزگذاری شده برای رمزگشایی: ")
                decoded = simple_decode(text)
                print(f"متن اصلی: {decoded}")

```

```
else:
    print("لطفاً عدد 1 تا 5 را وارد کنید.")

except ValueError:
    print("خطا: لطفاً عدد وارد کنید.")
except Exception as e:
    print(f"خطای غیرمنتظره: {e}")
```