```python
"""
CP1404/CP5632 - Practical - Suggested Solution
Password checker built from "skeleton" code to help you get started
"""

MIN_LENGTH = 2
MAX_LENGTH = 6
SPECIAL_CHARS_REQUIRED = False
SPECIAL_CHARACTERS = "!@#$%^&*()_-=+`~,./'[]<>?{}|\\"


def main():
    """Program to get and check a user's password."""
    print("Please enter a valid password")
    print("Your password must be between", MIN_LENGTH, "and", MAX_LENGTH,
          "characters, and contain:")
    print("\t1 or more uppercase characters")
    print("\t1 or more lowercase characters")
    print("\t1 or more numbers")
    if SPECIAL_CHARS_REQUIRED:
        print("\tand 1 or more special characters: ", SPECIAL_CHARACTERS)
    password = input("> ")
    while not is_valid_password(password):
        print("Invalid password!")
        password = input("> ")
    print("Your " + str(
        len(password)) + " character password is valid: " + password)


def is_valid_password(password):
    """Determine if the provided password is valid."""
    # TODO: if length is wrong, return False
    if len(password) < MIN_LENGTH or len(password) > MAX_LENGTH:
        return False

    count_lower = 0
    count_upper = 0
    count_digit = 0
    count_special = 0
    for char in password:
        # TODO: count each kind of character (use str methods like isdigit)
        if char.isdigit():
            count_digit += 1
        elif char.islower():
            count_lower += 1
        elif char.isupper():
            count_upper += 1
        elif char in SPECIAL_CHARACTERS:
            count_special += 1

    # TODO: if any of the 'normal' counts are zero, return False
    if count_lower == 0 or count_upper == 0 or count_digit == 0:
        return False

    # TODO: if special characters are required, then check the count of those
    # and return False if it's zero
    if SPECIAL_CHARS_REQUIRED:
        if count_special == 0:
            return False

    # if we get here (without returning False), then the password must be valid
    return True


main()
```