



HCMUTE

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

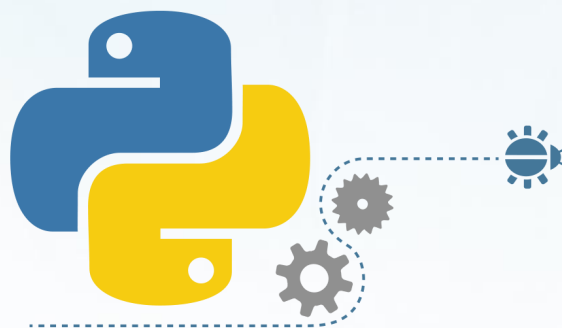
HCMC University of Technology and Education



KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN HỆ THỐNG THÔNG TIN

NHẬP MÔN LẬP TRÌNH PYTHON (IPPA233277)

BÀI TẬP XỬ LÝ CHUỖI



GV. Trần Quang Khải

1. Khái niệm và cấu trúc của chuỗi
2. Truy cập giá trị trong chuỗi
3. Định dạng chuỗi
4. Các ký tự thoát
5. Chuỗi dạng Unicode
6. Các hàm cơ bản trên chuỗi



Ví dụ: Viết một chương trình để kiểm tra xem một chuỗi có phải là chuỗi đối xứng hay không?

```
def is_palindrome(s):  
    # Chuyển đổi chuỗi thành chữ thường và loại bỏ các khoảng trắng  
    s = s.lower().replace(" ", "")  
    # So sánh chuỗi với chuỗi nghịch đảo của nó  
    return s == s[::-1]  
  
# Nhập chuỗi từ người dùng  
input_string = input("Nhập chuỗi cần kiểm tra: ")  
  
# Gọi hàm kiểm tra và in kết quả  
if is_palindrome(input_string):  
    print("Chuỗi là chuỗi đối xứng.")  
else:  
    print("Chuỗi không phải là chuỗi đối xứng.")
```



1. Viết chương trình để in ra tất cả các ký tự trong một xâu ký tự.
2. Viết chương trình để đảo ngược một xâu ký tự.
3. Viết chương trình để tìm số lần xuất hiện của một chuỗi con trong một xâu ký tự.
4. Viết chương trình để kiểm tra xem một xâu ký tự có phải là một chuỗi đối xứng hay không.
5. Viết chương trình để loại bỏ các ký tự trùng lặp trong một xâu ký tự.
6. Viết chương trình để chuyển đổi các ký tự viết hoa thành viết thường và ngược lại trong một xâu ký tự.
7. Viết chương trình để tìm độ dài của một xâu ký tự.
8. Viết chương trình để sắp xếp các ký tự trong một xâu ký tự theo thứ tự tăng dần.
9. Viết chương trình để tìm xâu con dài nhất không chứa các ký tự trùng lặp trong một xâu ký tự.
10. Viết chương trình để tìm ký tự xuất hiện nhiều lần nhất trong một xâu ký tự.
11. Viết chương trình để tìm xâu con có độ dài lớn nhất và không chứa các ký tự lặp lại trong một xâu ký tự.
12. Viết chương trình để tìm số lượng từ trong một xâu ký tự.



13. Viết chương trình để kiểm tra xem một chuỗi ký tự có phải là một đảo ngược của chuỗi ký tự khác hay không.
14. Viết chương trình để tạo một chuỗi ký tự mới bằng cách thay thế các ký tự trong một chuỗi ký tự bằng các ký tự khác.
15. Viết chương trình để tìm tất cả các chuỗi con của một chuỗi ký tự có độ dài lớn hơn hoặc bằng một giá trị cho trước.
16. Viết chương trình để tìm chuỗi con dài nhất chứa tất cả các ký tự trong một chuỗi ký tự.
17. Viết chương trình để tìm tất cả các chuỗi con khác nhau của một chuỗi ký tự.
18. Viết chương trình để tìm tất cả các chuỗi con lặp lại của một chuỗi ký tự.
19. Viết chương trình để chuyển đổi một chuỗi ký tự thành mã ASCII và ngược lại.
20. Viết chương trình để đếm số lượng ký tự số, ký tự chữ cái in hoa, ký tự chữ cái thường và các ký tự đặc biệt trong một chuỗi ký tự.



21. Viết chương trình cho người dùng nhập một chuỗi, sau đó hiển thị chuỗi đó ra màn hình theo chiều ngược lại

Ví dụ: Chuỗi: Tran Dinh Khang, xuất ra: gnahK hniD narT

22. Viết chương trình cho phép người dùng nhập một chuỗi ký tự số

- a. kiểm tra xem đây có phải là một dãy số Căn Cước hợp lệ hay không? Biết rằng, hợp lệ khi ký tự chữ số nằm trong $[0 \dots 9]$ và có độ dài 12 số?
- b. Xuất ra một số thông tin cơ bản gồm tỉnh – giới tính – năm sinh. Biết rằng cấu trúc số Căn Cước gồm:
 - ✓ 03 chữ số đầu là mã tỉnh
 - ✓ 01 chữ số tiếp theo là giới tính công dân
 - ✓ 02 chữ số tiếp theo là năm sinh
 - ✓ 06 chữ số cuối là khoảng số ngẫu nhiên định danh



23. Xây dựng chương trình cho người dùng nhập một địa chỉ thư điện tử từ bàn phím, kiểm tra xem địa chỉ thư điện tử này có hợp lệ hay không? Biết rằng, hợp lệ khi:

- ✓ Chứa đúng một ký tự '@'
- ✓ Có định dạng a@b.c trong đó a, b, c là các chuỗi khác rỗng

24. Đếm số lần xuất hiện của các từ trong một string (không phân biệt hoa thường).

Ví dụ: câu “An eye for an eye makes the whole world blind – Mahatma Gandhi” có 2 từ ‘an’, 2 từ ‘eye’, 1 từ ‘for’...

Gợi ý: Đổi string thành chữ thường (lower()) và dùng dictionary chứa các từ và số lần xuất hiện của chúng.

25. Kiểm tra một password có mạnh không, biết rằng password được xem là mạnh nếu nó có dài tối thiểu là 8 và có chứa: chữ cái, chữ số, ký tự đặc biệt (không phải chữ cái và chữ số, ví dụ #%@).

Ví dụ: 'SPKT1122@' là password mạnh; '123456789', 'spkttpchm' hoặc '#\$#&*^#*(\$' không phải là passwords mạnh.

26. Tạo password ngẫu nhiên là một password như mô tả trong câu trên.

Gợi ý: Tạo ra một số nguyên ngẫu nhiên nằm trong khoảng giá trị ASCII của chữ cái, chữ số, ký tự đặc biệt, rồi chuyển mã ASCII thành ký tự bằng hàm chr().



27. Tìm hiểu hàm `title()` và hàm `translate()`: nêu công dụng và viết code minh họa.

28. Kiểm tra xem một chuỗi có phải là chuỗi **symmetrical** hay không, một chuỗi gọi là symmetrical nếu hai nửa của chuỗi đều giống nhau.

Ví dụ:

- ✓ khokho là chuỗi symmetrical
- ✓ amaama là chuỗi symmetrical
- ✓ abcdab không là chuỗi symmetrical

29. Viết chương trình tối ưu chuỗi. Biết rằng, chuỗi tối ưu khi không chứa các khoảng trắng dư thừa, các từ cách nhau đúng bằng một khoảng trắng.

30. Viết một hàm đặt tên là `NegativeNumberInStrings(str)`. Hàm này có đối số truyền vào là một chuỗi bất kỳ, Hãy viết lệnh để xuất ra các số nguyên âm trong chuỗi.

Ví dụ: Nếu nhập vào chuỗi “abc-5xyz-12k9l--p” thì hàm phải xuất ra được 2 số nguyên âm đó là -5 và -12



31. Viết chương trình tối ưu chuỗi danh từ. Một chuỗi được gọi là tối ưu khi: không chứa các khoảng trắng dư thừa, các từ cách nhau bởi một khoảng trắng, ký tự đầu tiên của các từ Viết Hoa

Ví dụ: Input “ TRẦN duY thAnH ” → Output “Trần Duy Thanh”

32. Cho một chuỗi là đường dẫn của một tập tin định dạng bất kỳ. Ví dụ: “D:\\music\\muabui.mp3”

Hãy viết 02 hàm để:

- ✓ Lấy ra **muabui.mp3**

- ✓ Lấy ra **muabui**

Lưu ý đường dẫn bài hát là bất kỳ. Nên khi truyền vào bài hát nào thì lấy chính xác theo bài hát đó.



- ✓ Họ tên : **Trần Quang Khải**
- ✓ Email : **khaitq@hcmute.edu.vn**
- ✓ Zalo (mã Qr)

