1.1 Python Introduction Python là ngôn ngữ lập trình nào?

- Python là một ngôn ngữ lập trình tự do và mã nguồn mở.

Ai là tác giả chính của Python?

- Python được tạo ra bởi Guido van Rossum, một nhà phát triển ngôn ngữ lập trình người Hà Lan. Ông Guido van Rossum được coi là tác giả chính của Python và đã đóng vai trò quan trọng trong sự phát triển của ngôn ngữ này.

Python được phát triển vào năm nào?

- Python bắt đầu phát triển vào thập kỷ 1980 và phiên bản đầu tiên, Python 0.9.0, được phát hành vào tháng 2 năm 1991.

Python là ngôn ngữ lập trình thông dịch (interpreted) hay biên dịch (compiled)?

- Python là một ngôn ngữ lập trình thông dịch (interpreted). Điều này có nghĩa rằng mã nguồn Python được thực thi bằng cách dịch từ mã nguồn gốc thành mã bytecode và sau đó thực thi trực tiếp bởi máy ảo Python (Python virtual machine).

Python có ưu điểm gì nổi bật so với các ngôn ngữ lập trình khác?

Python có một số ưu điểm nổi bật so với các ngôn ngữ lập trình khác, bao gồm:

-Được coi là dễ học và dễ đọc, với cú pháp rõ ràng và đơn giản.

-Hỗ trợ nhiều thư viện và frameworks mạnh mẽ, giúp phát triển ứng dụng nhanh chóng.

- Đa năng và có thư viện phong phú, được sử dụng trong nhiều lĩnh vực như web development, data science, machine learning, và nhiều ứng dụng khác.

- Cộng đồng lớn và tích cực, cung cấp sự hỗ trợ và tài liệu đa dạng.

- Tương thích đa nền tảng, có sẵn cho nhiều hệ điều hành.

-Hỗ trợ cho nhiều paradigms lập trình như lập trình hướng đối tượng, lập trình hàm và lập trình thủ tục.

1.2. Python Applacation

Python được sử dụng trong những lĩnh vực ứng dụng nào trong phát triển web ?

- Python được sử dụng trong phát triển web trong các lĩnh vực sau:

- Phát triển các ứng dụng web động và tĩnh.

- Xây dựng các hệ thống quản lý nội dung (CMS).

- Phát triển ứng dụng e-commerce.-

- Tạo các ứng dụng web dựa trên dữ liệu và APIs.

- Xây dựng các ứng dụng web thời gian thực sử dụng các frameworks như Django và Flask.

Python được áp dụng trong các ngành công nghiệp nào khác ngoài công nghệ

thông tin và phát triển phần mềm ?

* Python được áp dụng trong các ngành công nghiệp bao gồm: tài chính, y tế, điện ảnh, hàng không, năng lượng, giáo dục, IoT, robotics, quản lý dự án, và nhiều lĩnh vực khác.

1.3. Python DataTypes

Kiểu dữ liệu nào được sử dụng để lưu trữ số nguyên trong Python? cho ví dụ.

-Kiểu dữ liệu được sử dụng để lưu trữ số nguyên trong Python là “int”.   
Ví dụ: x=10

Kiểu kiểu dữ liệu nào được sử dụng để lưu trữ danh sách các phần tử có thứ tự

trong Python? cho ví dụ.

-Kiểu dữ liệu được sử dụng để lưu trữ danh sách các phần tử có thứ tự trong Python là “List”

Ví dụ : list=[1,2,3,4,5]

Làm thế nào để tạo một tuple trong Python? cho ví dụ.

- Để tạo một tuple trong Python , ta sử dụng dấu ngoặc đơn hoặc dấu ngoặc đôi:

Ví dụ : tuple= (1,2,3) hoặc tuple= 1,2,3.

Kiểu dữ liệu nào được sử dụng để lưu trữ tập hợp các giá trị duy nhất trong

Python? cho ví dụ.

-Kiểu dữ liệu được sử dụng để lưu trữ tập hợp các giá trị duy nhất trong Python là set.  
 Ví dụ: my\_set = {1, 2, 3, 4, 5}

Làm thế nào để tạo một từ điển (dictionary) trong Python? cho ví dụ.

Để tạo một từ điển (dictionary) trong Python, bạn sử dụng cặp key-value và đặt chúng trong dấu ngoặc nhọn {}.   
Ví dụ: my\_dict = {'name': 'John', 'age': 30, 'city': 'New York'}

Kiểu dữ liệu nào được sử dụng để lưu trữ thời gian và ngày trong Python? Cho  
ví dụ.

Kiểu dữ liệu được sử dụng để lưu trữ thời gian và ngày trong Python là “datetime”. Để tạo một đối tượng datetime, bạn cần sử dụng thư viện datetime.

Ví dụ: from datetime import datetime

current\_time = datetime.now()

print(current\_time)

Kiểu dữ liệu nào được sử dụng để biểu diễn một ký tự đơn trong Python? cho ví

dụ.

Kiểu dữ liệu được sử dụng để biểu diễn một ký tự đơn trong Python là str (string).

Dưới đây là ví dụ về cách tạo một biến str để biểu diễn một ký tự đơn:

Mychar=’A’

Làm thế nào để kiểm tra kiểu dữ liệu của một biến trong Python? cho ví dụ.

Để kiểm tra kiểu dữ liệu của một biến trong Python, bạn có thể sử dụng hàm type(). Dưới đây là ví dụ cách sử dụng type() để kiểm tra kiểu dữ liệu của một biến:  
 my\_integer=42   
 typeofso=type(my\_integer)

Kiểu dữ liệu nào được sử dụng để biểu diễn các giá trị logic True và False trong

Python? cho ví dụ

Kiểu dữ liệu được sử dụng để biểu diễn các trị logic True và False trong Python là “bool”.  
Ví dụ : is\_true = True

is\_false = False

Làm thế nào để chuyển đổi một chuỗi thành một số nguyên trong Python? Cho ví dụ

Để chuyển đổi một chuỗi thành một số nguyên trong Python, bạn có thể sử dụng hàm int(). Dưới đây là ví dụ cách sử dụng int() để thực hiện chuyển đổi:

Ví dụ:

Mystring =”42”

MyInteger= int(Mystring)

Làm thế nào để tạo một danh sách rỗng trong Python? cho ví dụ.

Để tạo một danh sách sách rỗng trong Python chúng ta sử dụng dấu ngoặc vuông ‘[ ]’ hoặc sử dụng constructor của kiểu danh sách ‘list()’

Ví dụ :

Chuoirong=[]

Hoặc

Chuoirong=list()

Kiểu dữ liệu nào được sử dụng để biểu diễn giá trị không tồn tại hoặc trống

trong Python? cho ví dụ.

Kiểu dữ liệu được sử dụng để biểu diễn giá trị không tồn tại hoặc trống trong Python là NoneType. Giá trị duy nhất trong kiểu dữ liệu này là None.

Ví dụ:   
my\_variable = None

if my\_variable is None:

print("my\_variable không tồn tại.")

else:

print("my\_variable có giá trị:", my\_variable)

Làm thế nào để thêm một phần tử vào danh sách trong Python? cho ví dụ.

Để thêm một phần tử vào danh sách trong Python, ta sử dụng append()  
Ví dụ:

list=[1,2,3,4,5]

a = 6

a.append(list)

Kiểu dữ liệu nào được sử dụng để biểu diễn các giá trị nhị phân trong Python?

cho ví dụ

Kiểu dữ liệu được sử dụng để biểu diễn các giá trị nhị phân trong Python là bytes. Đối tượng bytes chứa một dãy các giá trị nhị phân không thể thay đổi.  
Ví dụ:  
binary\_data = bytes.fromhex('48656c6c6f2c20576f726c6421')

print(binary\_data) # Kết quả: b'Hello, World!'

Làm thế nào để kiểm tra xem một giá trị có thuộc một danh sách hay không

trong Python? cho ví dụ.

Bạn có thể kiểm tra xem một giá trị có thuộc một danh sách hay không bằng cách sử dụng toán tử in hoặc hàm in.  
Ví dụ:

my\_list = [1, 2, 3, 4, 5]

# Kiểm tra xem giá trị 3 có trong danh sách không

if 3 in my\_list:

print("3 thuộc danh sách.")

else:

print("3 không thuộc danh sách.")

Kiểu dữ liệu nào được sử dụng để biểu diễn các biểu thức điều kiện trong

Python? cho ví dụ.

Kiểu dữ liệu được sử dụng để biểu diễn các biểu thức điều kiện trong Python là bool (boolean). Biểu thức điều kiện trả về một giá trị boolean, có thể là True (đúng) hoặc False (sai).  
Ví dụ  
x = 5

y = 10

result = x > y

print("x > y:", result) # Kết quả: False

Làm thế nào để xóa một phần tử khỏi danh sách trong Python? cho ví dụ.

Để xóa một phần tử khỏi danh sách trong Python, bạn có thể sử dụng một số phương thức, bao gồm remove(), pop(), và del.  
Ví dụ:   
my\_list = [1, 2, 3, 4, 5]

my\_list.remove(3)

print(my\_list) # Kết quả: [1, 2, 4, 5]

1.3.1 Python List

List trong Python là gì và cách khai báo một List

List trong Python là một danh sách dùng để lưu trữ các giá trị có thể thay đổi.  
Các khai báo một List :   
numbers = [1, 2, 3, 4, 5]  
fruits = ["apple", "banana", "cherry"]  
mixed\_types = [1, "apple", 3.14, True]

Làm thế nào để truy cập và lấy giá trị của một phần tử trong một List.

Để truy cập và lấy giá trị của một phần tử trong một List thì bạn gọi tên list đó ra và   
và kẹp theo vị trí cả phần tử đó ở trong list bằng dấu [ ].  
Ví dụ : print(myList[0]). #In ra phần tử đầu tiên trong danh sách myList.

Python cho phép thực hiện các thao tác nào để thêm và xóa phần tử trong một List

Để thao tác thêm thì ta dùng append, và xóa thì ta dùng remove.  
Làm thế nào để tìm độ dài (số lượng phần tử) của một List

Ta dùng “len(“Tên list”) để lấy độ dài các phần tử trogn danh sách

Python cung cấp cách nào để lặp qua các phần tử của một List

Để lập các phần tử qua một List ta dùng vòng lặp for in

1.3.2 Python Set

Python Set là gì và cách khai báo một Set trong Python

Set trong Python là một cấu trúc dữ liệu lưu trữ tập hợp các phần tử duy nhất, không có thứ tự cụ thể, và sử dụng dấu ngoặc nhọn `{}` hoặc hàm `set()` để khai báo.

Cách khai báo:

+ sử dụng dấu `{ }`: myset={1,2,3,4}

+sử dung hàm set()= myset=set([1,2,3,4])

Làm thế nào để kiểm tra xem một phần tử có tồn tại trong một Set hay không trong Python

Sử dụng toán tử `in` để kiểm tra xem một phần tử có tồn tại trong Set hay không . Nếu có nó sẽ trả về `True` và `False` nếu không tồn tại.

Python Set có thể chứa các phần tử trùng lặp không? Nếu có, làm thế nào để xóa các phần tử trùng lặp trong Set

Trong Python Set không thể chứa các phần tử trùng lặp.  
Có thể xóa các phần tử trùng lặp trong Set bằng cách sau : Nếu bạn muốn tạo một bản sao của set mà không có phần tử trùng lặp, bạn có thể tạo một danh sách từ set ban đầu và sau đó chuyển danh sách đó lại thành một set.  
  
Ví dụ :

my\_set = {1, 2, 2, 3, 3, 3}  
unique\_list = list(my\_set)  
new\_set = set(unique\_list)  
print(new\_set) # Kết quả: {1, 2, 3}

Hãy nêu cách thực hiện các phép toán tập hợp cơ bản như hợp (union), giao (intersection) và hiệu (difference) trên các Set trong Python.

Hợp (Union): Hợp của hai set chứa tất cả các phần tử từ cả hai set, và loại bỏ các phần tử trùng lặp.  
Ví dụ:  
set1 = {1, 2, 3}

set2 = {3, 4, 5}

union\_set = set1.union(set2)

Giao (Intersection): Giao của hai set chứa các phần tử mà cả hai set đều chứa.  
 Ví dụ:

set1 = {1, 2, 3}

set2 = {3, 4, 5}

intersection\_set = set1.intersection(set2)

Hiệu (Difference): Hiệu của hai set chứa các phần tử có trong set đầu tiên nhưng không có trong set thứ hai.

Ví dụ:

set1 = {1, 2, 3}

set2 = {3, 4, 5}

difference\_set = set1.difference(set2)

Làm thế nào để thêm và xóa phần tử từ một Set trong Python

Có 2 phương thức cơ bản để thêm và xóa phần tử từ một Set trong Python.

Để thêm thì dùng `add()` còn để xóa thì dùng `remove()`

1.3.3. Python Puple

Python Tuple là gì và cách khai báo một Tuple trong Python?

Tuple là kiểu dữ liệu chuỗi bao gồm nhiều các thành phần có kiểu dữ liệu khác nhau, nhưng nó là không thể thay đổi được.

Cách khai báo : sử dụng dấu `{ }` để truyền các phần tử vào tuple.

Ví dụ: tuple\_toan\_so = (1, 2, 3)

Trong Python, Tuple có thể chứa các kiểu dữ liệu gì?  
`

Tuple trong Python là một kiểu dữ liệu dùng để lưu trữ các đối tượng có thứ tự và bất biến.

Làm thế nào để truy cập các phần tử trong một Tuple?

Sử dụng toán tử index [] để truy cập vào một phần tử của tuple.

Python Tuple có khác biệt gì so với Python List?

Tuple, đó là List có thể thay đổi (mutability) trong khi Tuple là bất biến (immutability).

Làm thế nào để thay đổi một phần tử trong một Tuple?

Chúng ta không thể thay đổi các phần tử trong một tuple. Điều này có nghĩa là, bạn không thể xóa hoặc loại bỏ các giá trị ra khỏi một tuple mà chỉ có thể xóa cả tuple đi mà thôi. Để xóa một hay nhiều tuple thì chúng ta sử dụng hàm del.

1.3.4. Python Dictionary

Python Dictionary là gì và có cấu trúc như thế nào

Kiểu dữ liệu Dictionary trong Python là một tập hợp các cặp key-value không có thứ tự, có thể thay đổi và lập chỉ mục (truy cập phần tử theo chỉ mục).

Làm thế nào để tạo một Dictionary trong Python và đưa các phần tử vào Dictionary

Để tạo một từ điển (dictionary) trong Python và đưa các phần tử vào từ điển, bạn có thể sử dụng cặp giá trị key-value và dấu hai chấm : để định nghĩa các phần tử của từ điển. Dưới đây là cách tạo và đưa các phần tử vào một từ điển:

Thêm phần tử vào dictionary python bằng phương thức setdefault()

mydict = {"NB":"Ninh Bình", "ND":"Nam Định", "TH":"Thanh Hóa"}

mydict.setdefault("TH","Thanh Hóa")

print(mydict)

Cách truy cập các giá trị trong một Dictionary trong Python

Lấy giá trị ( value ) của phần tử sử dụng khóa ( key ) đặt trong cặp dấu ngoặc vuông [] Lấy giá trị tương ứng với khóa bằng cách truyền khóa vào như tham số của phương thức get get()

Làm thế nào để thêm, sửa đổi hoặc xóa phần tử trong một Dictionary

Để cập nhật các item trong dictionary, hoặc cập nhật dictionary, ta có thể thêm một item mới vào dictionary, hoặc gán một value mới cho một key đã tồn tại.

Python Dictionary có những phương thức và tính năng gì giúp thực hiện các thao tác phổ biến

Truy cập và thay đổi phần tử:

dict[key]: Truy cập giá trị của phần tử có key tương ứng.

dict[key] = value: Thay đổi giá trị của phần tử hoặc thêm một phần tử mới nếu key chưa tồn tại.

Lấy danh sách các keys và giá trị:

dict.keys(): Trả về một danh sách các keys trong từ điển.

dict.values(): Trả về một danh sách các giá trị trong từ điển.

dict.items(): Trả về danh sách các cặp key-value trong từ điển dưới dạng tuple.

Kiểm tra sự tồn tại của key:

key in dict: Kiểm tra xem một key có tồn tại trong từ điển hay không

Xóa phần tử:

del dict[key]: Xóa phần tử có key tương ứng khỏi từ điển.

Lấy giá trị với mặc định (default value):

dict.get(key, default): Trả về giá trị của key nếu tồn tại, hoặc giá trị mặc định nếu key không tồn tại.

Xóa tất cả các phần tử:

dict.clear(): Xóa tất cả các phần tử khỏi từ điển, làm cho từ điển trở thành rỗng.

Sao chép từ điển:

dict.copy(): Tạo một bản sao của từ điển.

1.3.5 String

Làm thế nào để khai báo một chuỗi (string) trong Python

Chuỗi trong Python được đánh dấu bằng dấu nháy đơn ' hoặc nháy kép “, tuy nhiên nếu bắt đầu bằng dấu nào thì phải kết thúc bằng dấu đấy

Python cung cấp những phương thức nào để cắt và nối chuỗi

Thường người mới bắt đầu, chuyển sang từ các ngôn ngữ khác sử dung `+`, ngoài ra hiều người sẽ dùng join() để nối chuỗi.

Làm thế nào để kiểm tra độ dài của một chuỗi trong Python

Cú pháp sử dụng hàm len để tính độ dài chuỗi trong python

Python cho phép định dạng chuỗi như thế nào, ví dụ: làm thế nào để chuyển một số thành chuỗi có định dạng cụ thể.

Python cho phép định dạng chuỗi bằng cách sử dụng các ký tự đặc biệt được gọi là các chuỗi thoát (escape sequences) hoặc bằng cách sử dụng chuỗi định dạng (formatted string).

Làm thế nào để tìm kiếm và thay thế một phần của chuỗi trong Python

Hàm replace() trong Python trả về một bản sao của chuỗi ban đầu sau khi đã thay thế các chuỗi con cũ bằng chuỗi con mới..

Viết chương trình (hàm) tạo một xâu và in ra màn hình?

def create\_and\_print\_string():

my\_string = "Xin chào, đây là một ví dụ về tạo chuỗi và in ra màn hình."

print(my\_string)

create\_and\_print\_string()

Viết chương trình (hàm) tìm tần suất xuất hiện của một ký tự trong một xâu?

def tinhTanSuat(chuoi, kyTu):

dem = 0

for char in chuoi:

if char == kyTu:

dem += 1

return dem

chuoi = input()

kyTu = input()

tanSuat = tinhTanSuat(chuoi, kyTu)

print(f"Tần suất xuất hiện của ký tự '{kyTu}' trong chuỗi là: {tanSuat} lần.")

Viết chương trình (hàm) đảo ngược xâu?

def reverse\_string(s):

return s[::-1]

Viết chương trình (hàm) tìm xâu con dài nhất trong một xâu?

def longest\_substring(s):

n = len(s)

table = [[0] \* n for \_ in range(n)]

for i in range(n):

table[i][i] = 1

start, max\_len = 0, 1

for length in range(2, n+1):

for i in range(n-length+1):

j = i + length - 1

if s[i] == s[j] and table[i+1][j-1] == j-i-1:

table[i][j] = length

if length > max\_len:

start = i

max\_len = length

return s[start:start+max\_len]

Viết chương trình (hàm) tìm xâu con xuất hiện nhiều nhất trong một xâu?

def most\_frequent\_substring(s):

freq\_map = {}

n = len(s)

max\_count, max\_substring = 0, None

for i in range(n):

for j in range(i+1, n+1):

substring = s[i:j]

freq\_map[substring] = freq\_map.get(substring, 0) + 1

if freq\_map[substring] > max\_count:

max\_count = freq\_map[substring]

max\_substring = substring

return max\_substring

Viết chương trình (hàm) loại bỏ khoảng trắng đầu và cuối của một xâu?

def loaiBoKhoangTrang(chuoi):

chuoiDaLoaiBo = chuoi.strip()

return chuoiDaLoaiBo

chuoi = input("Nhập chuỗi: ")

chuoiLoaiBoKhoangTrang = loaiBoKhoangTrang(chuoi)

print("Chuỗi sau khi loại bỏ khoảng trắng đầu và cuối:", chuoiLoaiBoKhoangTrang).

Viết chương trình (hàm) chuyển một xâu sang viết hoa hoặc viết thường?

def chuyenChuoi(chuoi, luaChon):

if luaChon == "hoa":

return chuoi.upper()

elif luaChon == "thuong":

return chuoi.lower()

else:

return chuoi

chuoi = input("Nhập chuỗi: ")

luaChon = input("Chọn chế độ (hoa/thuong): ")

chuoiChuyenDoi = chuyenChuoi(chuoi, luaChon)

print("Chuỗi sau khi chuyển đổi:", chuoiChuyenDoi)

Viết chương trình (hàm) tách một xâu dựa trên ký tự hoặc xâu trắng?  
   
 def tachChuoi(chuoi, kyTuTach):

danhSachPhan = chuoi.split(kyTuTach)

return danhSachPhan

chuoi = input("Nhập chuỗi: ")

kyTuTach = input("Nhập ký tự hoặc xâu trắng dùng để tách: ")

danhSachPhan = tachChuoi(chuoi, kyTuTach)

print("Các phần sau khi tách chuỗi:", danhSachPhan)

Viết chương trình (hàm) tìm số lần xuất hiện của một xâu con trong một xâu khác?

def timSoLanXuatHien(chuoiLon, chuoiCon):

soLanXuatHien = chuoiLon.count(chuoiCon)

return soLanXuatHien

chuoiLon = input("Nhập chuỗi lớn: ")

chuoiCon = input("Nhập chuỗi con cần tìm: ")

soLanXuatHien = timSoLanXuatHien(chuoiLon, chuoiCon)

print(f"Số lần xuất hiện của chuỗi '{chuoiCon}' trong chuỗi '{chuoiLon}': {soLanXuatHien} lần.")

Viết chương trình (hàm) tìm số lần xuất hiện của một xâu con trong một xâu khác?

def timSoLanXuatHien(chuoiLon, chuoiCon):

soLanXuatHien = 0

i = 0

while i < len(chuoiLon):

if chuoiLon[i:i+len(chuoiCon)] == chuoiCon:

soLanXuatHien += 1

i += len(chuoiCon) - 1

i += 1

return soLanXuatHien

chuoiLon = input("Nhập chuỗi lớn: ")

chuoiCon = input("Nhập chuỗi con cần tìm: ")

soLanXuatHien = timSoLanXuatHien(chuoiLon, chuoiCon)

print(f"Số lần xuất hiện của chuỗi '{chuoiCon}' trong chuỗi '{chuoiLon}': {soLanXuatHien} lần.")

Viết chương trình (hàm) đếm số các kí tự là chữ cái tiếng Anh trong xâu S cho trước?

def demChuCaiTiengAnh(chuoi):

soLuongChuCai = 0

for char in chuoi:

if char.isalpha():

soLuongChuCai += 1

return soLuongChuCai

chuoi = input("Nhập chuỗi: ")

soLuongChuCaiTiengAnh = demChuCaiTiengAnh(chuoi)

print(f"Số lượng chữ cái tiếng Anh trong chuỗi: {soLuongChuCaiTiengAnh} ký tự.")

Cho trước hai xâu kí tự s1, s2. Viết chương trình (hàm) in ra xâu kí tự bao gồm lần lượt các kí tự được lấy ra từ s1, s2. Nếu một trong hai xâu s1, s2 hết trước thì lấy tiếp từ xâu còn lại. Ví dụ nếu s1 = “012”, s2 = “abcde” thì xâu kết quả sẽ là “0a1b2cde”

def noiHaiXau(s1, s2):

i = 0

j = 0

ketQua = ""

while i < len(s1) and j < len(s2):

ketQua += s1[i] + s2[j]

i += 1

j += 1

ketQua += s1[i:] + s2[j:]

return ketQua

s1 = input("Nhập xâu thứ nhất: ")

s2 = input("Nhập xâu thứ hai: ")

ketQua = noiHaiXau(s1, s2)

print("Xâu kết quả:", ketQua)

python

Cho trước hai xâu kí tự s1, s2. Viết chương trình (hàm) tính in ra xâu kí tự là phần chung đầu tiên cực đại của hai xâu s1, s2. Ví dụ nếu s1 = “abcdeghik”, s2 = “abcmnopq” thì xấu cần tính là “abc”.

def timPhanChungDauTien(s1, s2):

phanChung = ""

minLen = min(len(s1), len(s2))

for i in range(minLen):

if s1[i] == s2[i]:

phanChung += s1[i]

else:

break

return phanChung

s1 = input("Nhập xâu thứ nhất: ")

s2 = input("Nhập xâu thứ hai: ")

ketQua = timPhanChungDauTien(s1, s2)

print("Phần chung đầu tiên cực đại của hai xâu là:", ketQua)

Cho trước xâu kí tự S. Viết chương trình (hàm) xoá đi các dấu cách thừa trong xâu S. Dấu cách thừa là các dấu cách ở đầu, cuối và giữa các từ. Quy định giữa các từ chỉ có một dấu cách. Ví dụ nếu S = ” baby table cloud ” thì sau khi sửa S sẽ trở thành “baby table cloud”.

def xoaDauCachThua(chuoi):

chuoiXoaDauCach = " ".join(chuoi.split())

return chuoiXoaDauCach

chuoi = input("Nhập xâu: ")

chuoiDaSua = xoaDauCachThua(chuoi)

print("Xâu sau khi xoá các dấu cách thừa:", chuoiDaSua)

Viết chương trình (hàm) nhập một xâu là họ tên đầy đủ (gồm họ, đệm, tên) từ

bàn phím, sau đó in ra màn hình lần lượt họ, đệm và tên vừa nhập. Ví dụ:

Nhập họ tên đầy đủ: Nguyễn Xuân Quang Lâm

Họ: Nguyễn

Đệm: Xuân Quang

Tên: Lâm

def tachHoTen(hoTen):

danhSachTen = hoTen.split()

ho = danhSachTen[0]

ten = danhSachTen[-1]

dem = " ".join(danhSachTen[1:-1])

return ho, dem, ten

hoTenDayDu = input("Nhập họ tên đầy đủ: ")

ho, dem, ten = tachHoTen(hoTenDayDu)

print("Họ:", ho)

print("Đệm:", dem)

print("Tên:", ten)

Viết chương trình (hàm) nhập n số nguyên từ bàn phim, các số cách nhau bởi

dấu cách, sau đó in ra màn hình:

– Danh sách số đã nhập trên một hàng ngang.

– Số lớn nhất và chỉ số của số lớn nhất.

– Số nhỏ nhất và chỉ số của số nhỏ nhất.

def main():

# Nhập danh sách số từ người dùng

danhSachSo = input("Nhập các số nguyên cách nhau bằng dấu cách: ").split()

danhSachSo = [int(so) for so in danhSachSo]

# In danh sách số đã nhập

print("Danh sách số đã nhập:", danhSachSo)

# Tìm số lớn nhất và chỉ số của số lớn nhất

max\_so = max(danhSachSo)

index\_max = danhSachSo.index(max\_so)

print(f"Số lớn nhất là {max\_so} và nằm ở chỉ số {index\_max} trong danh sách.")

# Tìm số nhỏ nhất và chỉ số của số nhỏ nhất

min\_so = min(danhSachSo)

index\_min = danhSachSo.index(min\_so)

print(f"Số nhỏ nhất là {min\_so} và nằm ở chỉ số {index\_min} trong danh sách.")

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

main()

Cho trước xâu kí tự S và xâu substr. Viết chương trình (hàm) tính số lần lặp của

xâu con substr trong xâu S, cho phép chồng lấn của các xâu con này.

Ví dụ nếu S = “12121332121432121”, substr = “121” thì số lần lặp là 4

def tinhSoLanLap(S, substr):

soLanLap = 0

start = 0

while start < len(S):

index = S.find(substr, start)

if index == -1:

break

soLanLap += 1

start = index + 1

return soLanLap

S = input("Nhập xâu S: ")

substr = input("Nhập xâu substr: ")

soLanLap = tinhSoLanLap(S, substr)

print(f"Số lần lặp của xâu con '{substr}' trong xâu '{S}' là {soLanLap}.")