BÀI 7: List – Tuple – Set – Dictionary

Mục tiêu chính: Cung cấp cho học viên kiến thức và kỹ năng sử dụng:

- *List*
- *Tuple*
- *Set*
- *Dictionary*

7.1. Độ dài của các danh sách

✓ **Yêu cầu:**

– Cho 4 danh sách sau:

a = [1, 2, 3]

b = [1, [2, 3]]

c = []

d = [1, 2, 3][1:]

– Cho biết độ dài của các danh sách trên là bao nhiêu?

7.2. Chọn đội trưởng của đội tệ nhất

✓ **Mô tả:**

– Bạn đang phân tích các đội thể thao. Các thành viên của mỗi đội được lưu trữ trong một danh sách. Huấn luyện viên là cái tên đầu tiên trong danh sách, đội trưởng là cái tên thứ hai trong danh sách, và các cầu thủ khác được liệt kê sau đó. Các danh sách này được lưu trữ trong một danh sách khác, bắt đầu với nhóm tốt nhất và tiếp tục qua danh sách cho đến nhóm kém nhất cuối cùng.

✓ **Yêu cầu:**

– Viết lệnh để chọn ra đội trưởng của đội tệ nhất.

▪ **Nhập**:

– Danh sách các đội, thứ tự giảm dần từ đội tốt nhất

– Trong mỗi đội: đứng đầu danh sách là tên huấn luyện viên, kế đến là tên đội trưởng, sau cùng là tên các vận động viên

```
teams = [['Steven', 'Neena', 'Lex', 'Alexander', 'Bruce'], ['David', 'Jack', 'Bill', 'Tom', 'Mike', 'Daniel'],
    ['Alexander', 'Adam', 'Payam', 'Kevin', 'Sigal', 'Mike']]
```

▪ **Xuất**:

– Tên đội trưởng của đội tệ nhất

7.3. List of animal

✓ **Yêu cầu: Tìm thú trong vườn thú**

– Sử dụng shell

– Tạo ra một list có các con thú.

- Nhập vào một con thú cần tìm

  => Chương trình in ra danh sách các con thú, số lượng các con thú và kết quả tìm kiếm như hình dưới:

```
List of animals: ['ant', 'bear', 'cat', 'dog', 'elephant', 'fish', 'goat', 'hippo']
Number of animals: 8
I want to find:
bear
There is bear in list of animals
```

- **Nhập**:
  - Nhập list, con thú cần tìm
- **Xuất**:
  - List
  - Số lượng thú
  - Kết quả tìm kiếm

✓ **Hướng dẫn**

- Trong project Python_co_ban, tạo package **Bai7**
- Trong package Bai7, tạo module có tên là **list_of_animals.py**

7.4. List numbers 1

✓ **Yêu cầu: Viết chương trình xử lý list như sau**

- Tạo list
- Cho phép người dùng lần lượt nhập các phần tử số cho list cho đến khi không muốn nhập nữa
- Nhập vào một số x

  => Chương trình sẽ trả lời những câu hỏi sau:
- Tính tổng các phần tử trong list
- x có xuất hiện trong list hay không không? Nếu có thì cho biết x xuất hiện bao nhiêu lần?
- x có lớn hơn tất cả các số trong list không?
- Nếu không thì x nhỏ hơn những số nào trong list? (In ra tất cả các số lớn hơn x)

```
Nhập giá trị:   10
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   5
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   -2
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   23
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   5
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   6
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   7
Tiếp tục nhập giá trị? 1: Có, 0: không  0
Nhập giá trị cần tìm x: 5
List: [10, 5, -2, 23, 5, 6, 7]
Tổng các giá trị trong list: 54
5 xuất hiện 2 lần trong list
5 không lớn hơn tất cả các số trong list
Các số lớn hơn 5 trong list: [10, 23, 6, 7]
```

✓ **Tóm tắt yêu cầu**

- **Nhập**:
  - Giá trị của các phần tử trong list
  - x
- **Xuất**:
  - List
  - Tổng list
  - Kết quả tìm x trong list
  - Các số lớn hơn x trong list

✓ **Hướng dẫn**

- Trong package Bai7, tạo module có tên là **list_numbers_1.py**

7.5. List numbers 2

✓ **Yêu cầu: Viết chương trình thực hiện xử lý list như sau**

- Tạo list
- Nhập số phần tử trong list
- Cho phép người dùng lần lượt nhập các phần tử cho list cho đến khi không muốn nhập nữa

=> Chương trình sẽ thực hiện những công việc sau:

- Tìm và in ra tất cả các số nguyên tố có trong list
- Tính trung bình cộng của các phần tử âm/ phần tử dương trong list
- Tìm giá trị lớn nhất/ nhỏ nhất trong list
- Sắp xếp list theo giá trị tăng dần

```
Nhập giá trị:   1
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   -3
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   2
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   8
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   -4
Tiếp tục nhập giá trị? 1: Có, 0: không  1
Nhập giá trị:   7
Tiếp tục nhập giá trị? 1: Có, 0: không  0
List: [1, -3, 2, 8, -4, 7]
Các số nguyên tố trong list: [2, 7]
Các phần tử âm trong list: [-3, -4]
Trung bình cộng các phần tử âm: -3.50
Các phần tử dương trong list: [1, 2, 8, 7]
Trung bình cộng các phần tử dương: 4.50
Giá trị max trong list 8
Giá trị min trong list -4
List sắp tăng dần: [-4, -3, 1, 2, 7, 8]
```

- ✓ **Tóm tắt yêu cầu**
  - **Nhập**:
    - Giá trị của các phần tử trong list
  - **Xuất**:
    - List
    - Các số nguyên tố trong list
    - Trung bình cộng âm/ dương trong list
    - Max/ min trong list
    - List tăng dần
- ✓ **Hướng dẫn**
  - Trong package Bai7, tạo module có tên là **list_number_2.py**

7.6. List numbers 3

- ✓ **Yêu cầu: Viết chương trình thực hiện xử lý list như sau**
  - Tạo list với 10 chiều cao (đơn vị tính là inch) là: 74, 74, 72, 72, 73, 69, 69, 71, 76, 71
  - Đổi inch sang mét: 1 m = 0.0254 inch
  - In ra 3 chiều cao đầu tiên, 3 chiều cao cuối cùng
  - In ra chiều cao trung bình, chiều cao lớn nhất, chiều cao nhỏ nhất
  - Sắp xếp list theo giá trị tăng dần

- Sắp xếp list theo giá trị giảm dần

7.7. Lớn hơn một phần tử

✓ **Yêu cầu:**

- Viết phương thức ***elementwise_greater_than (L, thresh)*** trả về một danh sách có cùng độ dài với danh sách L, trong đó giá trị tại chỉ mục i là True nếu L[i] lớn hơn thresh, và nếu L[i] nhỏ hơn hoặc bằng thresh thì là False.

▪ **Nhập:**

```
L = [1, 2, 3, 4]
thresh = 2
```

▪ **Xuất**:

```
print(elementwise_greater_than(L, thresh)) # in ra [False, False, True, True]
```

7.8. Xác định trong danh sách các số có chứa "số may mắn" hay là không?

✓ **Yêu cầu:**

- Một danh sách các số được xem là "may mắn" nếu có ít nhất một số chia hết cho 7.
- Hãy viết phương thức **has_lucky_number(nums)** trả về xem danh sách các số đã cho có phải là "may mắn" hay không?

▪ **Nhập:**

```
nums = [2, 6, 7, 9]
```

▪ **Xuất**:

```
print(has_lucky_number(nums)) # in ra True
```

7.9. Xác định Khách có đến dự tiệc muộn hay không?

✓ **Mô tả:**

- Chúng ta sử dụng list để ghi lại những người đã tham dự bữa tiệc của chúng ta và họ đến theo thứ tự nào.
- Danh sách arrivals đại diện cho một bữa tiệc có 7 khách, trong đó Adela xuất hiện đầu tiên và Ford là người đến cuối cùng:
- arrivals = ['Adela', 'Fleda', 'Owen', 'May', 'Mona', 'Gilbert', 'Ford']
- Một vị khách được coi là 'trễ tiệc' nếu họ đến sau ít nhất một nửa số khách của bữa tiệc. Tuy nhiên, họ không phải là khách cuối cùng. Trong danh sách trên, Mona và Gilbert là những vị khách duy nhất đến trễ tiệc.

✓ **Yêu cầu: Viết phương thức party_late(arrivals, name) để xem liệu vị khách có tên đó có đến 'trễ tiệc' hay không?**

```
def party_late(arrivals, name):
    # trả về True nếu Khách đến trễ
    # trả về False nếu Khách không đến trễ
```

- **Xuất**:

```
print(party_late(arrivals, name = 'Gilbert'))
print(party_late(arrivals, name = 'Ford'))
print(party_late(arrivals, name = 'Mona'))
```

```
True
False
True
```

7.10. Xác định các bữa ăn liên tiếp có nhàm chán hay là không?

✓ **Yêu cầu:**

- Viết phương thức menu_is_boring(meals) đưa ra danh sách các bữa ăn được phục vụ trong một khoảng thời gian, trả về True nếu cùng một bữa ăn đã từng được phục vụ hai ngày liên tiếp và False nếu không.

- **Nhập:**

```
meals_1 = ['Redneck Ribs', 'Prawn Star', 'San Quentin Squid',
    'Fist Full of Pizza', 'Honky Tonk Chicken']

meals_2 = ['Redneck Ribs', 'Prawn Star', 'Running Bear Salmon',
    'Running Bear Salmon', 'Honky Tonk Chicken']
```

- **Xuất**:

```
print(menu_is_boring(meals_1)) # in False
print(menu_is_boring(meals_2)) # in True
```

7.11. Tuple strings

✓ **Yêu cầu: Viết chương trình thực hiện việc xử lý tuple như sau:**

- Tạo 1 tuple có 10 phần tử chuỗi bất kỳ.
- Nhập index dương (0<=index<10), index âm (-1 >=index>=-9)
- Nhập chuỗi cần tìm s_find

  => Chương trình sẽ thực hiện những công việc sau:
- In tuple
- In giá trị của phần tử trong tuple có index dương và index âm đã nhập
- Tìm và đếm số lần xuất hiện của s_find trong tuple

```
Tuple: ('red', 'green', 'yellow', 'blue', 'black', 'white', 'pink', 'orange', 'red', 'blue')
Nhập số từ 0 đến 9:     2
Nhập số từ -1 đến -9:   -2
Nhập chuỗi cần tìm:
blue
Tuple[ 2 ]= yellow
Tuple[ -2 ]= red
blue xuất hiện trong tuple 2 lần
```

✓ **Tóm tắt yêu cầu**

- **Nhập**:
  - Tuple, index dương, index âm
  - s_find
- **Xuất**:
  - Theo yêu cầu trên

✓ **Hướng dẫn**

- Trong package Bai7, tạo module có tên là **tuple_strings.py**

7.12. Tuple numbers

✓ **Yêu cầu: Viết chương trình thực hiện việc xử lý trên tuple như sau:**

- Tạo 1 tuple a chứa 4 số nguyên dương đầu tiên
- Tạp 1 tuple b chứa 4 số nguyên dương tiếp theo
- Tạo 1 tuple c là sự kết hợp của các phần tử trong tuple a và b
- Tạo 1 tuple d từ tuple c với các phần tử được sắp xếp
- In phần tử thứ 3 của d
- In 3 phần tử cuối cùng của d

```
Tuple a: (3, 1, 2, 4)
Tuple b: (5, 7, 6, 8)
Tuple c: (3, 1, 2, 4, 5, 7, 6, 8)
Tuple d: (1, 2, 3, 4, 5, 6, 7, 8)
Tuple[3]= 4
3 phần tử cuối cùng của tuple d (6, 7, 8)
```

✓ **Tóm tắt yêu cầu**

- **Nhập**:
  - Tuple a, b
- **Xuất**:
  - Tuple c, d
  - Phần tử thứ 3 của d
  - 3 phần tử cuối cùng của d

✓ **Hướng dẫn**

- Trong package Bai7, tạo module có tên là **tuple_numbers.py**

7.13. Set numbers

✓ **Yêu cầu: Viết chương trình thực hiện việc xử lý trên set như sau:**

- Khai báo và khởi tạo set1, set2
- Cho phép người dùng lần lượt nhập các phần tử số cho set1 cho đến khi không muốn nhập nữa
- Cho phép người dùng lần lượt nhập các phần tử số cho set2 cho đến khi không muốn nhập nữa

  => Chương trình sẽ thực hiện những công việc sau::
- In set1, set2
- Cho biết mỗi set có bao nhiêu phần tử, tổng giá trị các phần tử của mỗi set
- Tìm giá trị lớn nhất, nhỏ nhất của mỗi set
- Lấy ra một phần tử ở set1 và in ra phần từ này
- Thực hiện set union của set1 và set2 và in kết quả
- Thực hiện set intersection của set1 và set2 và in kết quả
- Thực hiện set difference của set1 với set2 và in kết quả
- Thực hiện set symmetric difference của set1 với set2 và in kết quả
- Sắp xếp set1 tăng dần và set2 giảm dần

```
Nhập giá trị cho element trong set 1:   5
Bạn có tiếp tục nhập set 1? 1: có, khác 1: không        1
Nhập giá trị cho element trong set 1:   2
Bạn có tiếp tục nhập set 1? 1: có, khác 1: không        1
Nhập giá trị cho element trong set 1:   7
Bạn có tiếp tục nhập set 1? 1: có, khác 1: không        1
Nhập giá trị cho element trong set 1:   4
Bạn có tiếp tục nhập set 1? 1: có, khác 1: không        0
Nhập giá trị cho element trong set 2:   7
Bạn có tiếp tục nhập set 2? 1: có, khác 1: không        1
Nhập giá trị cho element trong set 2:   8
Bạn có tiếp tục nhập set 2? 1: có, khác 1: không        1
Nhập giá trị cho element trong set 2:   10
Bạn có tiếp tục nhập set 2? 1: có, khác 1: không        1
Nhập giá trị cho element trong set 2:   6
Bạn có tiếp tục nhập set 2? 1: có, khác 1: không        0
Set 1: {2, 4, 5, 7}
Set 2: {8, 10, 6, 7}
Chiều dài Set 1: 4
Chiều dài Set 2: 4
Tổng Set 1: 18
Tổng Set 2: 31
Max Set 1, Min Set 1: 7 , 2
Max Set 2, Min set 2: 10 , 6
Pop Set 1: 2
Set 1 sau khi pop: {4, 5, 7}
Set 1 union Set 2: {4, 5, 6, 7, 8, 10}
Set 1 intersection Set 2: {7}
Set 1 difference Set 2: {4, 5}
Set 1, Set 2 symmetric differnce: {4, 5, 6, 8, 10}
Set 1 tăng dần: [4, 5, 7]
Set 2 giảm dần: [10, 8, 7, 6]
```

- ✓ **Tóm tắt yêu cầu**
  - **Nhập**:
    - set1, set2
  - **Xuất**:
    - Theo các yêu cầu liệt kê ở trên
- ✓ **Hướng dẫn**
  - Trong package Bai7, tạo module có tên là **set_numbers.py**

7.14. Danh bạ điện thoại

- ✓ **Yêu cầu: Viết chương trình thực hiện việc xử lý danh bạ điện thoại như sau:**

- Tạo một danh bạ kiểu dictionary để lưu trữ danh bạ điện thoại với các cặp key-value, ví dụ như:

Name	Telephone number
Johnny	0989741258
Katherine	0903852147
Misu	0913753951
Jack	0933753654
…	…

0903852147

- Nhập tên cần tìm search_name
- Nhập tên, số điện thoại

  => Chương trình sẽ thực hiện những công việc sau:

- Tìm search_name trong danh bạ. Nếu không tìm thấy thì in thông tin tên – số điện thoại. Nếu không tìm thấy thì thông báo là không tìm thấy.
- Thêm một liên hệ mới với thông tin: tên – số điện thoại đã nhập
- In danh bạ

```
Bạn muốn làm gì? 1: Xem danh bạ; 2: Tìm kiếm, 3: Thêm mới       1
Danh bạ điện thoại:
Tên      Số điện thoại
Jack  --  0933753654
Misu  --  0913753951
Johnny  --  0989741258
Katherine  --  0903852147
Tiếp tục lựa chọn? 1: Có; 0: Không      1
Bạn muốn làm gì? 1: Xem danh bạ; 2: Tìm kiếm, 3: Thêm mới       2
Nhập tên cần tìm:
Misu
Misu có số điện thoại là: 0913753951
Tiếp tục lựa chọn? 1: Có; 0: Không      1
Bạn muốn làm gì? 1: Xem danh bạ; 2: Tìm kiếm, 3: Thêm mới       3
Nhập tên:
Louisa
Nhập số điện thoại:
0913852258
Danh bạ điện thoại:
Tên      Số điện thoại
Jack  --  0933753654
Misu  --  0913753951
Johnny  --  0989741258
Louisa  --  0913852258
Katherine  --  0903852147
Tiếp tục lựa chọn? 1: Có; 0: Không       0
```

- ✓ **Tóm tắt yêu cầu**
  - **Nhập**:
    - Danh bạ
    - search_name
    - Liên hệ mới
  - **Xuất**:
    - Kết quả tìm kiếm
    - Danh bạ
- ✓ **Hướng dẫn**
  - Trong package Bai7, tạo module có tên là **danh_ba.py**

7.15. Từ điển

- ✓ **Yêu cầu: Viết chương trình thực hiện việc xử lý từ điển Anh – Việt như sau:**
  - Tạo một từ điển

    => Chương trình sẽ thực hiện những công việc sau:

- Thêm từ vào từ điển (key: từ tiếng Anh, value: nghĩa tiếng Việt)
- Hiển thị từ điển, cho biết trong từ điển hiện tại có bao nhiêu từ
- Tìm kiếm từ tiếng Anh => nếu tìm thấy thì hiển thị key và value. Nếu không tìm thấy thì thông báo không tìm thấy
- Xóa một từ trong từ điển, dựa trên key cung cấp

```
Bạn muốn làm gì? 1: Xem từ điển; 2: Tra từ, 3: Thêm từ, 4: Xóa từ       1
Dictionary:
Từ Anh   Nghĩa Việt
cat       con mèo
dog       con chó
ant       con kiến
bear      con gấu
Tiếp tục lựa chọn? 1: Có; 0: Không      1
Bạn muốn làm gì? 1: Xem từ điển; 2: Tra từ, 3: Thêm từ, 4: Xóa từ       2
Nhập từ cần tra:        book
Không tìm thấy từ trong từ điển
Tiếp tục lựa chọn? 1: Có; 0: Không      1
Bạn muốn làm gì? 1: Xem từ điển; 2: Tra từ, 3: Thêm từ, 4: Xóa từ       3
Nhập từ Anh:    book
Nhập nghĩa Việt:        quyển sách
Dictionary:
Từ Anh   Nghĩa Việt
cat       con mèo
dog       con chó
book      quyển sách
ant       con kiến
bear      con gấu
Tiếp tục lựa chọn? 1: Có; 0: Không      1
Bạn muốn làm gì? 1: Xem từ điển; 2: Tra từ, 3: Thêm từ, 4: Xóa từ       4
Nhập từ cần xóa:        ant
Bạn có thật sự muốn xóa hay không? 1: Xóa, 0: Không     1
Đã xóa từ trong từ điển
Dictionary:
Từ Anh   Nghĩa Việt
cat       con mèo
dog       con chó
book      quyển sách
bear      con gấu
Tiếp tục lựa chọn? 1: Có; 0: Không      0
```

✓ **Tóm tắt yêu cầu**

- **Nhập**:
  - Từ cần tìm/ Từ cần thêm/ Từ cần xóa
- **Xuất**:
  - Kết quả như hình trên

✓ **Hướng dẫn**

- Trong package Bai7, tạo module có tên là tu_dien.py