

BÀI 3**THỰC HÀNH VỀ TỆP, MẢNG VÀ DANH SÁCH**

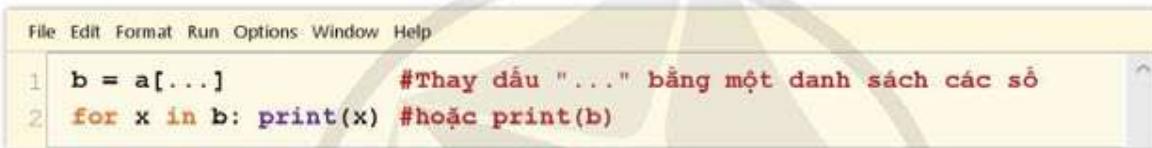
Học xong bài này, em sẽ:

- ✓ Sử dụng được lát cắt để xử lý mảng, danh sách theo ý muốn.
- ✓ Biết và sử dụng được một số hàm xử lý tệp dữ liệu đầu vào, đầu ra.

Nhiệm vụ 1. Lát cắt

Yêu cầu:

- a) Cho a là mảng (danh sách) các số. Hãy dùng lát cắt tạo danh sách b và dùng vòng lặp `for` in kết quả ra màn hình (xem mẫu ở *Hình 1*) để kiểm tra kết quả trong mỗi trường hợp sau:



```
File Edit Format Run Options Window Help
1 b = a[...]           #Thay dấu "..." bằng một danh sách các số
2 for x in b: print(x) #hoặc print(b)
```

Hình 1. Mẫu xử lý dùng vòng lặp `for`

- b là nửa cuối của a .
- b là một phần tử kể từ đầu trái của a .
- b là các phần tử chỉ số lẻ của a .

- b) Cho a là ma trận (bảng số) hình vuông $n \times n$ các số thực. Hãy viết các câu lệnh (dùng lát cắt khi có thể) để in kết quả ra màn hình và kiểm tra kết quả trong mỗi trường hợp sau:

- Các hàng chỉ số chẵn của a .
- Hai phần tử đầu tiên của hàng đầu tiên của a .
- Hai cột đầu tiên của a .
- Các cột chỉ số lẻ của a .

Gợi ý:

Toán tử lát cắt (*Hình 2*) trích ra *đoạn con* liền mạch hay *dãy con* (có bước nhảy `step` cách quãng) từ một dãy tuần tự nhiều mục dữ liệu, ví dụ như một biến kiểu danh sách.

Lát cắt	Kết quả
<code>a[lo:hi]</code>	Cắt đoạn con, từ chỉ số <code>lo</code> đến chỉ số <code>hi-1</code> .
<code>a[:hi]</code>	Cắt đoạn đầu danh sách (mặc định <code>lo=0</code>) đến chỉ số <code>hi-1</code> .
<code>a[lo:]</code>	Cắt đoạn cuối danh sách từ vị trí <code>lo</code> đến cuối danh sách.
<code>a[lo:hi:step]</code>	Cắt danh sách con, từ chỉ số <code>lo</code> đến chỉ số <code>hi-1</code> , với bước nhảy <code>step</code> .

Hình 2. Các lát cắt

Lưu ý: Nếu bước nhảy **step** nhận giá trị âm thì toán tử lát cắt sẽ đảo chiều, di từ cuối danh sách lên đầu danh sách, từ phải sang trái, kết quả nhận được giống như dùng phương thức **reverse()** (xem ví dụ *Hình 3*).

```
File Edit Format Run Options Window Help
1 ds = [1, 2, 3, 4, 5]
2 dsLui = ds[::-1] #dsLui = [5, 4, 3, 2, 1]
```

Hình 3. Ví dụ về lát cắt

Nhiệm vụ 2. Vòng lặp

Yêu cầu: Cho a là mảng hai chiều hình vuông gồm n hàng và n cột các số thực. Hãy tính:

- Tổng các phần tử chỉ số chẵn ở hàng i của a .
- Tổng các phần tử âm; tổng các phần tử không âm ở hàng i của a .
- In ra chỉ số các phần tử bằng số x cho trước.

Gợi ý: Vòng lặp **for** hoặc **while** duyệt qua các phần tử trong danh sách a và thân vòng lặp có thể xử lý lần lượt tất cả các phần tử hoặc chọn một số phần tử thỏa mãn điều kiện nào đó: theo chỉ số i hoặc theo giá trị $a[i]$.

Nhiệm vụ 3. Đọc dữ liệu từ tệp đầu vào và viết ra tệp

Yêu cầu:

Cho tệp “bangDiem.txt” gồm nhiều dòng; các mục dữ liệu cách nhau khoảng trắng:

- Dòng thứ nhất: Hai số nguyên dương n và m ; với n là số học sinh, m là số môn học.
- Dòng thứ hai: *TênHS Toán Văn Tin Li...* gồm $(m + 1)$ từ.
- n dòng tiếp theo, mỗi dòng có tên học sinh và điểm các môn học của học sinh đó.

Hãy viết một hàm **nhapTuTep()** để đọc tệp dữ liệu đầu vào “bangDiem.txt” và khởi tạo dữ liệu sẵn sàng để tính toán phân tích kết quả học tập:

- Một mảng hai chiều $n \times m$ các số thực.
- Hai danh sách: danh sách tên học sinh và danh sách tên môn học.

Hướng dẫn thực hiện:

Có thể tạo tệp “bangDiem.txt” bằng cách chỉnh sửa và bổ sung bảng trong *Hình 1a* ở Bài 2; từ Word hay Excel, thao tác **Copy\Paste** vào cửa sổ của Notepad hay cửa sổ soạn thảo của Python; ghi lưu thành tệp có định dạng text.

- Đọc từng dòng của tệp đầu vào.
- Chuyển đổi mỗi mục của danh sách sang kiểu dữ liệu cần thiết và nối thêm vào danh sách tương ứng trong chương trình (tham khảo chương trình ở *Hình 4*).

```

1 fi = open("bangDiem.txt", encoding = "utf-8")
2 dong = fi.readlines().split()
3 ...
4 for i in range(n):
5 ...
6     hang = [float(d) for d in dong]
7     a.append(hang) #Thêm một hàng vào mảng a[][]
```

Hình 4. Ví dụ câu lệnh thêm một hàng vào mảng hai chiều

Trong Python, nếu một dòng gồm nhiều mục khác kiểu dữ liệu, xen kẽ nhau, thì phải truy cập từng phần tử của danh sách và chuyển từ xâu kí tự thành kiểu dữ liệu đúng mô tả.

Các thao tác với tệp dữ liệu

Đầu vào là tệp thuần văn bản chữ và số (đuôi tên tệp “txt”) gồm nhiều dòng; mỗi dòng gồm nhiều từ, mỗi từ là một mục dữ liệu, phân cách bằng khoảng trắng.

Các bước thực hiện như sau:

Bước 1. Mở tệp để đọc hay viết, sử dụng hàm `open()` như ví dụ ở *Hình 5*.

```

1 f = open(name, mode) #name = đường dẫn và tên tệp
2 #mode = 'r' là mở để đọc (dây là trường hợp mặc định);
3 #mode = 'w' là mở để viết vào; mode = 'a' là mở để viết nối thêm
```

Hình 5. Cách mở tệp để đọc, viết và viết nối thêm

Bước 2. Đọc từ tệp, có thể dùng các phương thức `read()`, `readline()`, `readlines()` kết hợp với `split()`:

`read().split()`: Đọc từng từ và nối liên toàn bộ các dòng thành một danh sách các từ. Sử dụng khi tệp ngắn và cần xử lý toàn bộ nội dung tệp.

`readline().split()`: Đọc một dòng, trả về danh sách các từ, thường dùng nhất.

`readlines()`: Đọc toàn bộ tệp, trả về danh sách các dòng, mỗi dòng là một xâu kí tự, kết thúc bằng ‘\n’ (dấu xuống dòng).

Bước 3. Xuất ra tệp thuần văn bản: có thể dùng hàm `print`, sau khi đã chuyển đầu ra chuẩn từ màn hình sang tệp đã mở để viết vào như sau:

`sys.stdout = f #f: tên biến tệp đã mở để viết vào`

Bước 4. Đóng tệp, dùng phương thức `close()`.

Lưu ý: Nếu giữa các từ được phân cách nhau bằng dấu phẩy thì ta có tệp kiểu “csv” (*comma separated value*) và để cắt ra từng từ cần dùng `split(',')` thay vì dạng mặc định `split()`.

Đọc và viết tệp văn bản có chữ tiếng Việt

Một số kí tự trong tiếng Việt không có trong bảng mã ASCII. Để đọc, viết tệp văn bản có chữ tiếng Việt hay các văn bản nói chung, có dùng các kí tự không phải ASCII thì cần thêm tham số tuỳ chọn `encoding = 'utf-8'` vào các hàm xử lí tệp.



Tạo dãy số thực ngẫu nhiên; sử dụng các hàm `mean`, `median`, `mode` trong module `statistics` để:

- Tìm `mean` của a và đếm số phần tử bé hơn, bằng, lớn hơn `mean`.
- Tìm `median` của a và cho biết đó là phần tử nào hay nó ở giữa hai phần tử nào.
- Tìm `mode` của a và cho biết số lần xuất hiện và dãy các chỉ số tương ứng.
- Áp dụng để phân tích dãy điểm từng môn học của lớp 11A.

BÀI TÌM HIỂU THÊM

LIỆT KÊ MỘT SỐ PHƯƠNG THỨC TRONG PYTHON

Python có sẵn nhiều phương thức áp dụng cho kiểu danh sách. *Bảng 1* dưới đây chọn liệt kê một số phương thức chưa được thực hành nhiều. Kí hiệu `ds` là tên của một biến kiểu danh sách.

Bảng 1. Một số phương thức xử lí danh sách thường dùng

Phương thức	Kết quả
<code>ds.clear()</code>	Xoá rỗng <code>ds</code>
<code>ds.copy()</code>	Tạo bản sao của <code>ds</code>
<code>ds.reverse()</code>	Dảo ngược vị trí các phần tử trong <code>ds</code>
<code>ds.sort()</code>	Sắp xếp <code>ds</code> theo thứ tự tăng dần
<code>ds.append(x)</code>	Thêm một phần tử <code>x</code> vào cuối <code>ds</code>
<code>ds.extend(ds_2)</code>	Nối thêm danh sách <code>ds_2</code> vào cuối <code>ds</code>
<code>ds.index(x)</code>	Tìm vị trí xuất hiện của phần tử <code>x</code> trong <code>ds</code>
<code>ds.count(x)</code>	Đếm số lần xuất hiện phần tử <code>x</code> trong <code>ds</code>
<code>ds.insert(i, x)</code>	Chèn thêm <code>x</code> thành phần tử chỉ số <code>i</code> trong <code>ds</code>
<code>ds.pop(i)</code>	Gỡ bỏ phần tử chỉ số <code>i</code>
<code>ds.clear(i)</code>	Xoá các phần tử bắt đầu từ vị trí <code>i</code> đến hết <code>ds</code>
<code>ds.remove(x)</code>	Xoá phần tử <code>x</code> gặp đầu tiên trong <code>ds</code>