

# Trường Đại học Kinh tế - Đại học Huế

## Khoa Hệ thống thông tin Kinh tế

### Lập trình hướng đối tượng

### Bài tập 1 : Lập trình thủ tục

*Hạn cuối nộp bài : 12h00 thứ năm, ngày 07 tháng 03 năm 2013*

*Nộp bằng cách gửi email về địa chỉ [lvman@hce.edu.vn](mailto:lvman@hce.edu.vn)*

*Hãy đọc Quy định nộp bài tập*

## Giới thiệu

Trong bài tập này, các bạn sẽ vận dụng tất cả các khái niệm đã học trong phần đầu môn Lập trình hướng đối tượng và sử dụng ngôn ngữ C# để thiết kế, xây dựng một chương trình đơn giản cho phép xử lý số liệu thống kê.

## Xử lý số liệu bằng thống kê toán học

Thống kê toán học là môn khoa học suy luận, khái quát về toàn bộ tổng thể dựa trên thông tin của một phần trong toàn bộ tổng thể đó. Thống kê toán nhằm tìm ra những quy luật, những giá trị chưa biết của các hiện tượng, biến lượng ngẫu nhiên có tính chất số lớn, căn cứ trên số liệu thống kê thu thập được. Vì vậy, thống kê toán được áp dụng trong hầu hết tất cả các lĩnh vực kinh tế - xã hội, liên quan đến các vấn đề cả định lượng và định tính.

*Trong phần này sẽ nêu tóm tắt những lý thuyết cần thiết cho việc lập trình, các em tham khảo tài liệu môn Xác suất & Thống kê để hiểu rõ hơn.*

## Số liệu mẫu

Giả sử chúng ta cần nghiên cứu biến ngẫu nhiên  $X$  nào đó. Ta gọi  $F(x)$ ,  $\mu$ ,  $\sigma^2$  là hàm phân phối, kỳ vọng và phương sai của biến ngẫu nhiên  $X$ . Nói chung, chúng đều chưa được biết và chúng ta cần phải đi tìm. Ta gọi đó là các đặc trưng lý thuyết.

Thông tin duy nhất mà nhà thống kê toán học dựa vào để phân tích xử lý và rút ra các kết luận là  $n$  kết quả quan sát được mà ta thường gọi là một mẫu ngẫu nhiên. Mẫu này có thể cho dưới dạng bình thường :

$$x_1, x_2, \dots, x_n$$

Hoặc mẫu có thể được cho dưới dạng thu gọn :

$$x_1, x_2, \dots, x_k$$

$$m_1, m_2, \dots, m_k \text{ với } \sum_{i=1}^k m_i = n \text{ cỡ mẫu}$$

Trong đó :

$m_i$  là số lần mẫu nhận giá trị  $x_i$

$k$  gọi là cỡ mẫu thu gọn.

Nếu  $n$  giá trị mẫu đều khác nhau thì  $m_i = 1$  và  $k = n$

Hoặc mẫu có thể được cho dưới dạng khoảng  $(a_i, a_{i+1})$ , có  $m_i$  giá trị mẫu rơi vào khoảng đó :

Khoảng giá trị	$(a_1, a_2)$	$(a_2, a_3)$	...	$(a_k, a_{k+1})$
Số quan sát rơi vào	$m_1$	$m_2$	...	$m_k$

$$\sum_{i=1}^k m_i = n$$

Xuất phát từ  $n$  quan sát thu được này, chúng ta có thể giải quyết một số bài toán Thống kê quen thuộc nhưng hay được dùng trong các vấn đề thực tế thuộc nhiều lĩnh vực khác nhau mà những bài toán này đòi hỏi khối lượng tính toán lớn, cần sự trợ giúp đặc lực và có hiệu quả của máy tính.

## Hai đặc trưng mẫu

Nếu mẫu được cho dưới dạng thu gọn thì kỳ vọng mẫu được tính theo công thức :

$$\bar{X} = \frac{1}{n} \sum_{i=1}^k m_i * x_i$$

và phương sai mẫu :

$$s^2 = \frac{1}{n} \sum_{i=1}^k m_i * x_i^2 - \bar{X}^2$$

hoặc

$$\hat{s}^2 = \frac{n}{n-1} s^2$$

Nếu mẫu được cho dưới dạng khoảng  $(a_i, a_{i+1})$  thì ta sẽ chọn điểm đại diện

$$x_i = \frac{a_i + a_{i+1}}{2}$$

sau đó tính kỳ vọng và phương sai như trường hợp trên.

## Ước lượng giá trị trung bình

Trong thống kê ta đã biết rõ  $\bar{X}$  là ước lượng điểm tốt cho  $\mu$  và  $s^2$  hoặc  $\hat{s}^2$  là ước lượng tốt cho  $\sigma^2$ . Ngoài ra,  $\bar{X}$  và  $s^2$  còn được dùng trong nhiều bài toán thống kê khác.

Khi đó :

- Ước lượng điểm cho giá trị trung bình của đại lượng  $X$  là  $\bar{X}$
- Ước lượng khoảng (khoảng tin cậy) với độ tin cậy  $(1 - \alpha)$  cho giá trị  $X$  là :
  - Nếu  $\sigma^2$  đã biết (giả thiết cần có hoặc  $X$  phân phối chuẩn hoặc cỡ mẫu  $n \geq 30$ )

$$(\bar{X} - u_{\alpha/2} \frac{\sigma}{\sqrt{n}}; \bar{X} + u_{\alpha/2} \frac{\sigma}{\sqrt{n}})$$

- Nếu  $\sigma^2$  chưa biết (giả thiết cần có  $X$  phân phối chuẩn)

$$(\bar{X} - t_{\alpha/2}^{(n-1)} \frac{s}{\sqrt{n-1}}; \bar{X} + t_{\alpha/2}^{(n-1)} \frac{s}{\sqrt{n-1}})$$

## Kiểm định giá trị trung bình của tổng thể phân phối chuẩn

Nếu biến ngẫu nhiên gốc trong tổng thể phân phối chuẩn  $X \sim N(\mu, \sigma^2)$  với  $\mu$  chưa biết, khi đó  $m = \mu$ , trung bình tổng thể sẽ dùng kí hiệu  $\mu$ . Kiểm định giả thiết về tham số  $\mu$ , so sánh với một số thực  $\mu_0$  cho trước.

- Trường hợp đã biết phương sai tổng thể  $\sigma^2$

Với mức ý nghĩa  $\alpha$  cho trước, ta có tiêu chuẩn kiểm định :

$$U = \frac{(\bar{X} - \mu_0)\sqrt{n}}{\sigma}$$

- Với cặp giả thiết  $\begin{cases} H_0 : & \mu = \mu_0 \\ H_1 : & \mu \neq \mu_0 \end{cases}$

Miền bác bỏ  $H_0$  là  $W_\alpha = (-\infty; -u_{\alpha/2}) \cup (u_{\alpha/2}; +\infty)$

$$- \text{ Với cặp giả thiết } \begin{cases} H_0 : & \mu = \mu_0 \\ H_1 : & \mu > \mu_0 \end{cases}$$

Miền bác bỏ  $H_0$  là  $W_\alpha = (u_\alpha; +\infty)$

$$- \text{ Với cặp giả thiết } \begin{cases} H_0 : & \mu = \mu_0 \\ H_1 : & \mu < \mu_0 \end{cases}$$

Miền bác bỏ  $H_0$  là  $W_\alpha = (-\infty; -u_{\alpha/2})$

Nếu  $U \in W_\alpha$  thì bác bỏ giả thiết  $H_0$ , chấp nhận giả thiết  $H_1$ .

Ngược lại, thì chưa đủ cơ sở để bác bỏ giả thiết  $H_0$ .

- Trường hợp không biết phương sai tổng thể  $\sigma^2$

Với mức ý nghĩa  $\alpha$  cho trước, ta có tiêu chuẩn kiểm định :

$$T = \frac{(\bar{X} - \mu_0)\sqrt{n}}{S}$$

$$- \text{ Với cặp giả thiết } \begin{cases} H_0 : & \mu = \mu_0 \\ H_1 : & \mu \neq \mu_0 \end{cases}$$

Miền bác bỏ  $H_0$  là  $W_\alpha = (-\infty; -t_{\alpha/2}^{(n-1)}) \cup (t_{\alpha/2}^{(n-1)}; +\infty)$

$$- \text{ Với cặp giả thiết } \begin{cases} H_0 : & \mu = \mu_0 \\ H_1 : & \mu > \mu_0 \end{cases}$$

Miền bác bỏ  $H_0$  là  $W_\alpha = (t_{\alpha/2}^{(n-1)}; +\infty)$

$$- \text{ Với cặp giả thiết } \begin{cases} H_0 : & \mu = \mu_0 \\ H_1 : & \mu < \mu_0 \end{cases}$$

Miền bác bỏ  $H_0$  là  $W_\alpha = (-\infty; -t_{\alpha/2}^{(n-1)})$

Nếu  $T \in W_\alpha$  thì bác bỏ giả thiết  $H_0$ , chấp nhận giả thiết  $H_1$ .

Ngược lại, thì chưa đủ cơ sở để bác bỏ giả thiết  $H_0$ .

## Yêu cầu của bài tập

Hãy sử dụng tất cả các kiến thức về ngôn ngữ lập trình C# đã được học để lập trình tạo ra một ứng dụng xử lý số liệu thống kê đơn giản. Chương trình này cho phép người sử dụng nhập vào  $n$  kết quả quan sát *định lượng* của một mẫu ngẫu nhiên, rồi tính toán những đặc trưng lý thuyết như kỳ vọng, phương sai mẫu, ước lượng giá trị trung bình của tổng thể và kiểm định giả thuyết trung bình tổng thể.

Các yêu cầu cụ thể của chương trình như sau :

1. Xây dựng cấu trúc dữ liệu để lưu trữ  $n$  quan sát của mẫu ngẫu nhiên
2. Cho phép nhập  $n$  quan sát đó theo dạng thông thường
3. Cho phép nhập  $n$  quan sát đó theo dạng thu gọn
4. Cho phép nhập  $n$  quan sát đó theo dạng khoảng
5. Tính hai đặc trưng : kỳ vọng và phương sai mẫu
6. Ước lượng giá trị trung bình của tổng thể
7. Kiểm định giả thuyết trung bình tổng thể phân phối chuẩn
8. In các kết quả ra màn hình
9. Hỗ trợ đối số dòng lệnh
10. (Tuỳ chọn - cộng điểm) Cho phép nhập  $n$  quan sát của mẫu ngẫu nhiên theo 3 dạng từ file văn bản
11. (Tuỳ chọn - cộng điểm) Xuất kết quả tính toán được ra file văn bản

*Chú ý :*

1. Nên tổ chức chương trình dưới dạng hàm để dễ kiểm soát và sửa lỗi
2. Nên code từng phần một, sau mỗi phần phải kiểm tra kỹ càng đoạn code vừa được viết là hoạt động tốt rồi mới chuyển sang phần khác. *Ví dụ : sau khi code phần cho phép nhập dữ liệu vào, thì nên code đoạn code in dữ liệu ra để kiểm soát việc nhập dữ liệu vào là đúng*
3. Khi kiểm tra thì nên kiểm tra với lượng ít số liệu để dễ kiểm soát chương trình. Sau khi chương trình đã đúng với số ít dữ liệu đó thì mới chạy chương trình với nhiều dữ liệu hơn

## Test

Các bạn dùng 3 bộ test sau đây để kiểm tra chương trình và chụp ảnh màn hình kết quả vào báo cáo.

### Data 1

Theo dõi thời gian hoàn thành sản phẩm (phút) ở một số công nhân, ta có số liệu sau :

8	8.6	9.5	10	9.4	9.3	8.4	8.7	10.5	8.5	11	10.6
10.3	9	10.4	8.3	10.7	10.9	8.9	11	9.8	8.9	8	9.9
10	8.7	10.2	10.8	10.1	8.8	9.9	8.2	10.7	10.4	10.4	10

1. Tính hai đặc trưng mẫu
2. Ước lượng khoảng cho giá trị trung bình tổng thể với độ tin cậy 90% và phương sai chưa biết
3. Kiểm định trung bình tổng thể có bằng 10 phút hay khác 10 phút với mức ý nghĩa 10% và phương sai chưa biết

### Data 2

Tiến hành cân thử một số gói mì, người ta có kết quả như sau :

Trọng lượng (gam)	197	198	199	200	201
Số gói	20	15	17	33	15

1. Tính hai đặc trưng mẫu
2. Ước lượng khoảng cho giá trị trung bình tổng thể với độ tin cậy 95% với phương sai tổng thể bằng phương sai mẫu
3. Kiểm định trung bình tổng thể có lớn hơn 199 với mức ý nghĩa 10% và phương sai tổng thể bằng phương sai mẫu

### Data 3

Điều tra mức chi phí của một số sinh viên, người ta có kết quả như sau :

Chi phí (triệu/tháng)	1.2 - 1.3	1.3 - 1.4	1.4 - 1.6	1.6 - 1.8	1.8 - 2
Số sinh viên	6	9	8	10	8

1. Tính hai đặc trưng mẫu
2. Ước lượng khoảng cho giá trị trung bình tổng thể với độ tin cậy 80% với phương sai tổng thể là chưa biết
3. Kiểm định trung bình tổng thể lớn hơn 1.4 triệu đồng một tháng, với mức ý nghĩa 5% và phương sai tổng thể chưa biết

## Hướng dẫn lập trình

Những thuật toán, cấu trúc dữ liệu và cách xử lý được đưa ra trong các phần sau đây chỉ là **đề nghị**. Các bạn có thể đưa ra thuật toán, cấu trúc dữ liệu và cách xử lý riêng và hãy chỉ ra trong báo cáo của bạn. Thuật toán, cấu trúc dữ liệu và cách xử lý của bạn là tốt hơn sẽ được đánh giá cao hơn và ngược lại, điểm của bạn sẽ thấp nếu chương trình của bạn hỗ trợ ít chức năng hơn hoặc cách xử lý ít phức tạp hơn.

Trong mỗi chức năng các bạn nên xây dựng thành các hàm chức năng nhỏ hơn. Như vậy, chương trình sẽ không bị rối và dễ gỡ lỗi. Đồng thời các chức năng khác có thể sử dụng lại các hàm đó.

Các phần dưới đây sẽ được trình bày tuần tự theo một quy trình mà nếu các bạn làm theo từng bước thì có thể hoàn thành tốt chương trình. Các bước sau sẽ là phát triển tiếp của bước trước, nên không làm bước trước thì sẽ khó hoàn thành được bước sau.

## Cấu trúc dữ liệu để lưu trữ $n$ quan sát của mẫu ngẫu nhiên

Cấu trúc dữ liệu là một trong hai thành phần cốt lõi để tạo nên một chương trình máy tính và nó có tính quyết định đến việc lựa chọn thuật toán, cách xử lý trong chương trình. Do đó, khi phân tích thiết kế chương trình, ta nên lựa chọn cấu trúc dữ liệu tốt để việc xử lý sau này sẽ đơn giản và thuận tiện hơn.

Đầu tiên, chúng ta cần phân tích xem chương trình cần lưu những dữ liệu gì. Theo yêu cầu của bài toán, chúng ta cần lưu trữ  $n$  quan sát. Mỗi quan sát là một giá trị định lượng nào đó, ví dụ : tuổi, mức thu nhập hoặc điểm trung bình học phần,... Nhưng chương trình phải cho phép nhập liệu theo 3 dạng : dạng thường, dạng thu gọn và dạng khoảng. Như vậy, cấu trúc dữ liệu phải thoả mãn hai đặc điểm :

1. cho phép lưu trữ một loạt các số liệu

2. mỗi thành phần trong loạt số liệu đó là :

- một số với dạng thường (giá trị quan sát  $x_i$ )
- hai số với dạng thu gọn (giá trị quan sát  $x_i$  - số lần xuất hiện của giá trị quan sát đó trong  $n$  quan sát  $m_i$ )
- và ba số với dạng khoảng (giá trị đầu khoảng  $a_i$  - giá trị cuối khoảng  $a_{i+1}$  - số lần xuất hiện của các giá trị quan sát trong khoảng đó  $m_i$ )

Với đặc điểm đầu tiên, bạn có thể dùng mảng, **List**, **Stack** hoặc **Queue** để lưu trữ một loạt các số liệu. Mảng rõ ràng là đơn giản nhất, nhưng mảng lại không có tính linh động khi chúng ta cần mở rộng kích thước của mảng. **List** thì linh động và mạnh mẽ hơn. **Stack** và **Queue** thì có

thể dùng nhưng hai cấu trúc này phù hợp cho các bài toán khác hơn là để lưu một danh sách dữ liệu.

Với đặc điểm thứ hai, mảng một chiều hoặc **List** sẽ không thể nào phù hợp để lưu dữ liệu theo dạng nhập liệu thứ hai và thứ ba. Do đó, ta có thể sử dụng mảng hai chiều, mảng ba chiều, hoặc 2 và 3 mảng một chiều, hoặc 2 và 3 **List**. Ví dụ, với trường hợp sử dụng mảng ba chiều, có thể tạo ra một mảng ba chiều với kích thước  $[n, 3]$ . Trong đó,  $n$  là số dòng và 3 là số cột. Có thể quy định cột thứ nhất là để lưu các giá trị quan sát  $x_i$ , cột thứ ba là để lưu số lượng  $m_i$ . Riêng cột thứ hai chỉ được dùng cho trường hợp nhập liệu dạng khoảng, khi đó cột thứ nhất sẽ lưu giá trị đầu khoảng  $a_i$ , cột thứ hai sẽ lưu giá trị cuối khoảng  $a_{i+1}$ .

Ngoài ra, bạn có thể sử dụng mảng một chiều hoặc **List** mà mỗi thành phần là một kiểu cấu trúc (**struct**). Mỗi cấu trúc nên được định nghĩa tốt để lưu trữ mỗi quan sát. Ví dụ, chúng ta có thể định nghĩa một cấu trúc như sau :

```
1 struct Item
2 {
3     public double x; // gia tri quan sat, hoac gia tri dau khoang
4     public double y; // gia tri cuoi khoang
5     public int m; // so luong quan sat co cung gia tri nay
6 };
```

## Nhập dữ liệu

Phần này trình bày việc nhập dữ liệu cho cả ba dạng : thông thường, thu gọn và dạng khoảng. Ở đây, tôi sử dụng thuật ngữ thành phần cho một dữ liệu đầy đủ của một quan sát (tùy theo dạng) được nhập. Lúc đó, theo dạng thông thường, mỗi thành phần sẽ là mỗi giá trị  $x_i$ . Theo dạng thu gọn, mỗi thành phần sẽ là mỗi cặp  $(x_i, m_i)$ . Còn dạng khoảng thì mỗi thành phần là mỗi nhóm ba giá trị  $(a_i, a_{i+1}, m_i)$ . Như vậy, số lượng thành phần sẽ là số lần người sử dụng nhập dữ liệu quan sát vào chương trình.

Trong việc nhập dữ liệu, bạn có thể áp dụng theo hai kiểu sau :

### 1. Biết trước số lượng thành phần được nhập

Tức là, bạn sẽ yêu cầu người sử dụng nhập vào số lượng thành phần muốn nhập trước. Sau đó, dựa vào giá trị số lượng thành phần được nhập đó, bạn có thể dùng một vòng lặp để cho phép người sử dụng nhập vào từng thành phần một.

### 2. Không biết trước số lượng thành phần được nhập

Tức là, số lượng thành phần không được biết trước, nên sau khi người sử dụng nhập xong dữ liệu của một thành phần nào đó thì chương trình sẽ hỏi xem người sử dụng có muốn nhập tiếp hay không. Tùy vào câu trả lời của người sử dụng mà chương trình cho phép người sử dụng tiếp tục nhập hoặc thoát ra khỏi tiến trình nhập này.



Hoặc một cách khác, các bạn có thể dừng việc nhập khi người sử dụng nhập vào ký tự kết thúc file (end of file - Ctrl + Z). Xem ví dụ vòng lặp `while` ở slide số 16, chủ đề 7.

Sau đoạn chương trình cho phép nhập dữ liệu, bạn nên có đoạn chương trình cho phép in ra dữ liệu đã được nhập vào để người sử dụng kiểm tra lại dữ liệu mà họ đã nhập.

Các bạn có thể mở rộng chương trình hơn để hỗ trợ những tính năng sau :

- Nhập dữ liệu mới
- Thêm thành phần mới
- Xoá thành phần đã có
- Tìm kiếm thành phần nào đó

## Tính kỳ vọng và phương sai mẫu

Sau khi dữ liệu các thành phần đã được nhập xong, ta có thể cho chương trình tính và in ra kỳ vọng và phương sai mẫu theo công thức có sẵn.

## Ước lượng giá trị trung bình

Thuật toán đoạn chương trình ước lượng giá trị trung bình như sau :

Hỏi người sử dụng muốn thực hiện ước lượng điểm hay ước lượng khoảng  
Nếu là ước lượng điểm

In ra kết quả của ước lượng điểm theo công thức

Ngược lại

Hỏi độ tin cậy  $\alpha$  và phương sai tổng thể  $\sigma^2$

Nếu đã biết phương sai tổng thể  $\sigma^2$

Tính các giá trị của khoảng ước lượng theo công thức

In ra khoảng ước lượng của giá trị trung bình

Ngược lại

Tính các giá trị của khoảng ước lượng theo công thức

In ra khoảng ước lượng của giá trị trung bình

*Khi yêu cầu nhập phương sai tổng thể, các bạn có thể cho phép người sử dụng sử dụng lại giá trị phương sai mẫu  $s^2$  đã được tính để làm giá trị phương sai cho tổng thể.*

## Kiểm định giá trị trung bình của tổng thể phân phối chuẩn

Thuật toán đoạn chương trình kiểm định giá trị trung bình như sau :

Nhập giá trị trung bình tổng thể cần kiểm định

Nhập kiểu giả thiết  $H_1$  và mức ý nghĩa  $\alpha$

Nhập phương sai tổng thể  $\sigma^2$

Nếu phương sai tổng thể được nhập

Tính  $U$  theo công thức

Tuỳ theo kiểu giả thiết  $H_1$ , thực hiện kiểm tra  $U$  so với  $W_\alpha$

Nếu  $U \in W_\alpha$  thì in ra bác bỏ giả thiết  $H_0$ , chấp nhận giả thiết  $H_1$

Ngược lại, in ra chưa đủ cơ sở để bác bỏ giả thiết  $H_0$

Ngược lại

Tính  $T$  theo công thức

Tuỳ theo kiểu giả thiết  $H_1$ , thực hiện kiểm tra  $T$  so với  $W_\alpha$

Nếu  $T \in W_\alpha$  thì in ra bác bỏ giả thiết  $H_0$ , chấp nhận giả thiết  $H_1$

Ngược lại, in ra chưa đủ cơ sở để bác bỏ giả thiết  $H_0$

*Khi yêu cầu nhập phương sai tổng thể, các bạn có thể cho phép người sử dụng sử dụng lại giá trị phương sai mẫu  $s^2$  đã được tính để làm giá trị phương sai cho tổng thể.*

## Dữ liệu cho $u_{\alpha/2}$ và $t_{\alpha/2}^{(n-1)}$

Trong công thức tính  $U$  và  $T$  ở phần ước lượng và kiểm định, các bạn có các giá trị  $u_{\alpha/2}$  và  $t_{\alpha/2}^{(n-1)}$ . Các bạn có thể yêu cầu người sử dụng nhập vào các giá trị đó hoặc tự tổ chức trong chương trình một cấu trúc dữ liệu lưu trữ các giá trị đó. Khi cần thì chỉ cần lấy giá trị tương ứng ra để tính toán.

## Menu chương trình

Trong các chương trình có nhiều chức năng, thì người lập trình thường hỗ trợ một menu chương trình. Người sử dụng chỉ việc chọn một số hoặc một chữ cái để thực hiện một chức năng nào đó. Khi một chức năng nào đó đã thực hiện xong thì menu đó lại được in ra lại và chương trình hỏi người sử dụng tiếp tục muốn thực hiện chức năng nào. Chương trình chỉ kết thúc/dừng khi người sử dụng chọn số hoặc chữ cái tương đương chức năng dừng chương trình.

Ví dụ menu chương trình có thể như sau khi chương trình chạy :

CHUONG TRINH XU LY SO LIEU

0 - Nhap du lieu

1 - Xem du lieu

2 - Tinh hai dac trung

3 - Uoc luong gia tri trung binh

4 - Kiem dinh gia tri trung binh

5 - Thoat

Ban chon so may : \_

Để hỗ trợ tính năng này, người ta thường dùng vòng lặp `while`. Và khi người sử dụng nhập số là 5 (theo ví dụ trên) thì mới ngắt vòng lặp `while`. Xem slide 23, chủ đề 5 để biết cách tạo một menu như trên.

## Xử lý đối số dòng lệnh

Xem slide 38-40, chủ đề 7.

Với chương trình này, các bạn có thể hỗ trợ 4 dạng đối số dòng lệnh sau :

```
1 da.exe
2 da.exe <num>
3 da.exe <path>
4 da.exe <path1> <path2>
5 da.exe -h
```

Với `da.exe` là tên chương trình, `da` là viết tắt của `data analysis`.

Dạng thứ nhất chỉ gọi chương trình mà không đưa vào đối số nào cả. Như vậy, chương trình sẽ in ra menu chương trình và chờ người sử dụng lựa chọn chức năng tiếp theo để thực hiện.

Dạng thứ hai gọi chương trình kèm với một số (`<num>`). Số này thể hiện dạng nhập dữ liệu. Nếu là 0 - nhập liệu dạng thông thường, 1 - nhập liệu dạng thu gọn, 2 - nhập liệu dạng khoảng. Chương trình sẽ tùy vào `<num>` để thực hiện chức năng nhập tương ứng.

Dạng thứ ba là gọi chương trình kèm theo đường dẫn (`<path>`) đến tập tin văn bản chứa dữ liệu. (Đọc thêm phần Nhập dữ liệu từ file văn bản bên dưới)

Dạng thứ tư là gọi chương trình kèm theo hai đường dẫn (`<path1>`, `<path2>`). Đường dẫn thứ nhất là chứa dữ liệu nhập vào chương trình, đường dẫn thứ hai là tập tin để ghi dữ liệu ra. (Đọc thêm phần Nhập và xuất dữ liệu ở bên dưới)

Dạng thứ tư kèm theo đối số `-h` để yêu cầu in ra hướng dẫn sử dụng của chương trình.

Thuật toán của phần xử lý đối số dòng lệnh như sau :

Kiểm tra số lượng đối số dòng lệnh

Nếu lớn hơn 3 thì thông báo lỗi và in ra hướng dẫn sử dụng chương trình

Nếu không có đối số nào thì chạy chương trình bình thường

Nếu có 2 đối số thì chuyển đến việc nhập và xuất dữ liệu với file

Nếu có 1 đối số

Đối số đó là chuỗi `-h` (so sánh chuỗi ký tự trong đối số với chuỗi `"-h"`) thì in ra hướng dẫn sử dụng

Nếu thuộc dạng 2 (thử đổi chuỗi thành số) thì tùy vào số đối được cho thực hiện nhập liệu theo dạng đó

Nếu thuộc dạng 3 thì chuyển đến nhập dữ liệu từ file văn bản

**Chú ý :** Nếu bạn không làm chức năng thứ 10 và 11 thì có thể bỏ qua việc xử lý dạng thứ 3 và 4 của đối số dòng lệnh

## Nhập dữ liệu từ file văn bản

Nội dung của tập tin chứa dữ liệu thống kê có thể được quy định như sau :

- Dòng đầu tiên trong tập tin là 1 số - thể hiện một trong ba dạng : 0 - thông thường, 1 - thu gọn và 2 - khoảng
- Dòng thứ hai là 1 con số khác - là số thành phần (gọi là  $k$ )
- Từ dòng thứ ba đến hết file, có tất cả là  $k$  dòng, mỗi dòng có 2 hoặc 3 số tùy thuộc vào dạng số liệu
  - Dạng thông thường : cặp số  $x_i, 1$
  - Dạng thu gọn : cặp số  $x_i, m_i$
  - Dạng khoảng : ba giá trị  $a_i, a_{i+1}, m_i$

Ví dụ sau đây là nội dung của tập tin data3.txt :

```
1 2
2 5
3 1.2 1.3 6
4 1.3 1.4 9
5 1.4 1.6 8
6 1.6 1.8 10
7 1.8 2 8
```

Thuật toán đọc dữ liệu từ file sử dụng mảng 2 chiều với kích thước  $[n, 3]$  trong ví dụ của phần mô tả cấu trúc dữ liệu như sau :

```
Đọc dòng đầu tiên trong file để biết dạng dữ liệu
Đọc dòng thứ hai để lấy số thành phần (k)
Khởi tạo mảng với kích thước [k,3]
Duyệt qua các dòng của mảng
    Đọc dòng tiếp theo trong file
    Sử dụng hàm Split của kiểu string để tách ra các chuỗi con
        Nếu số thành phần không phải là 2 hoặc 3
            và không phù hợp với dạng nhập dữ liệu
                Huỷ khởi tạo mảng
                Đóng file, không đọc nữa
        Nếu là dạng nhập liệu thông thường hoặc thu gọn
            Gán giá trị của chuỗi con thứ nhất vào cột 1
            của dòng hiện tại trên mảng
            Gán giá trị của chuỗi con thứ hai vào cột 3
```

của dòng hiện tại trên mảng  
 Nếu dạng nhập liệu là khoảng  
 Gán giá trị của chuỗi con thứ nhất vào cột 1  
 của dòng hiện tại trên mảng  
 Gán giá trị của chuỗi con thứ hai vào cột 2  
 của dòng hiện tại trên mảng  
 Gán giá trị của chuỗi con thứ ba vào cột 3  
 của dòng hiện tại trên mảng

## Xuất dữ liệu ra file văn bản

Tất cả các dữ liệu tính toán được đều được xuất ra tập tin văn bản và theo khuôn dạng sau :

- Dòng đầu tiên là hai số của kỳ vọng và phương sai cách nhau bằng khoảng trống
  - Nếu có tính ước lượng thì hai dòng tiếp theo sẽ là kết quả của ước lượng
    - Dòng đầu của hai dòng này là các giả thiết của ước lượng
      - \* Nếu là ước lượng điểm thì chỉ ghi là `uld`
      - \* Nếu là ước lượng khoảng thì ghi theo dạng : `ulk <độ tin cậy  $\alpha$ > <phương sai  $\sigma^2$  (nếu có)>`
    - Ví dụ :
 

```
ulk 0.05
ulk 0.1 1.44
```
  - Dòng thứ hai là kết quả của ước lượng
    - \* Nếu là ước lượng điểm thì chỉ có một số tương ứng với kỳ vọng của tổng thể
    - \* Nếu là ước lượng khoảng thì là hai giá trị cách nhau bằng khoảng trống
- Còn nếu có tính kiểm định thì 4 dòng tiếp theo được dành cho các kết quả của kiểm định
  - Dòng thứ nhất là các giả thiết của kiểm định. Có thể theo khuôn dạng sau : `<kiểu giả thiết  $H_1$ > <mức ý nghĩa  $\alpha$ > <phương sai  $\sigma^2$  (nếu có)>`
  - Dòng thứ hai là giá trị U và T tính được
  - Dòng thứ ba là hai giá trị  $-u_{\alpha/2}$  và  $u_{\alpha/2}$  hoặc chỉ một giá trị  $u_\alpha$  hoặc  $-u_\alpha$
  - Cuối cùng là kết luận : Bác bỏ giả thiết  $H_0$ , chấp nhận giả thiết  $H_1$  hoặc chưa đủ cơ sở để bác bỏ giả thiết  $H_0$