

SAU BÀI HỌC NÀY EM SẼ:

- Thực hành thiết kế một số chương trình hoàn chỉnh theo mô đun.



Trong Bài 28, em đã biết tầm quan trọng và ý nghĩa của việc thiết kế chương trình theo mô đun. Mỗi mô đun chính là một chương trình con được xác định khi phân tích thiết kế bài toán lớn. Mỗi mô đun được viết một cách độc lập, có thể sử dụng lại như các thư viện và có thể chia sẻ trong nhóm làm việc. Trong bài thực hành này em sẽ được tự mình thiết kế chương trình cho một bài toán hoàn chỉnh theo mô đun.



### Nhiệm vụ. Tính điểm tổng hợp của vận động viên

Trong một cuộc thi đấu thể thao khu vực dạng Sea Games, các cầu thủ tham gia thi đấu ở các bộ môn khác nhau và do sự phức tạp của truyền tin nên ban tổ chức (BTC) đã quyết định nhập thông tin kết quả thi đấu của các vận động viên vào các tệp văn bản, sau đó các tệp này được chuyển về trung tâm để xử lí. Mỗi bộ môn thể thao sẽ có số lượng ban giám khảo khác nhau. Mỗi tệp dữ liệu lưu thông tin sẽ có khuôn dạng gồm nhiều dòng, mỗi dòng là thông tin của một vận động viên thi đấu tại một bộ môn nào đó. Mỗi dòng của tệp thông tin sẽ bao gồm:

- Đầu tiên là mã số của vận động viên.
- Tiếp theo là điểm cho của các giám khảo, giữa các điểm số đều có dấu cách.
- Điểm số được cho có thể là số nguyên hoặc số thập phân, thang điểm 10.

Theo quy định của BTC, quy định tính điểm chung cho mỗi vận động viên như sau: Điểm tổng hợp của mỗi vận động viên là trung bình cộng điểm của ban giám khảo. Tuy nhiên trong mỗi ban giám khảo luôn có hai giám khảo đặc biệt, hai giám khảo này có hệ số tổng hợp là 2, trong khi các giám khảo khác có hệ số 1. Theo quy định của BTC thì các giám khảo đặc biệt sẽ ở vị trí đầu tiên và cuối cùng của danh sách. Ở tất cả các bộ môn thể thao số lượng thành viên ban giám khảo phải lớn hơn 2.

Nhiệm vụ của em là tính điểm tổng hợp của các vận động viên từ tệp văn bản [SeaGames.inp](http://SeaGames.inp). Kết quả ghi ra tệp [ketqua.out](http://ketqua.out) có dạng như sau:

- Tệp sẽ có nhiều dòng, số dòng bằng đúng số dòng của tệp đầu vào.
- Trên mỗi dòng ghi mã của vận động viên và điểm tổng hợp tương ứng. Yêu cầu ghi điểm tổng hợp với hai chữ số sau dấu phẩy.

SeaGames.inp	
101	7.5 8.0 9.0 9.5 7.1 6.8
200	8.5 9.1 9.5 8.6 9.9
003	6.6 7.0 7.5 6.8 5.9 8.1
045	8.5 7.9 9.3 9.0 8.9
901	9.2 9.7 8.6

Ketqua.out	
101	7.77
200	9.14
003	7.08
045	8.71
901	9.06

Hướng dẫn:

### a) Phân tích tổng quát

Từ yêu cầu của bài toán, dễ thấy chương trình có thể chia thành ba công việc, hay mô đun con, tương đối độc lập như sau:

- Mô đun 1: Đọc dữ liệu từ tệp.
- Mô đun 2: Xử lý dữ liệu đã đọc ở mô đun 1.
- Mô đun 3: Ghi dữ liệu đã xử lý ra tệp theo yêu cầu.

Các mô đun sẽ được thiết kế một cách độc lập. Chương trình chính sẽ tổng hợp và kết nối các mô đun.

### b) Thiết kế mô đun nhập dữ liệu

Dữ liệu sẽ được nhập từ tệp [SeaGames.inp](#) và được đưa vào hai mảng:

- Mảng thứ nhất là danh sách mã số của các vận động viên.
- Mảng thứ hai có ý nghĩa như sau: Tương ứng với mỗi vận động viên từ mảng thứ nhất, mỗi phần tử của mảng thứ hai là dãy các điểm số mà ban giám khảo chấm cho vận động viên này.

Hai mảng này sẽ được đặt tên lần lượt là **DS** và **Diem**.

Mô đun này được thiết lập bằng hàm **nhapDL()** với tham số đầu vào là tên tệp dữ liệu. Hàm sẽ trả lại hai mảng **DS** và **Diem**. Chú ý: Khi đọc dữ liệu từ tệp, với mỗi dòng, phần tử đầu tiên chính là mã số của vận động viên, sẽ được đưa vào mảng **DS**, dãy các giá trị còn lại chính là dãy điểm và được đưa vào mảng **Diem**.

Hàm **nhapDL(fin)** được mô tả như sau:

---

```

1 def nhapDL(fin):
2     f = open(fin)
3     A = []
4     B = []
5     for line in f:
6         S = line.split()
7         A.append(S[0])
8         temp = S[1:len(S)]
9         temp = [float(x) for x in temp]
10        B.append(temp)
11    f.close()
12    return A,B

```

---



### c) Thiết kế mô đun xử lý dữ liệu chính

Mục đích của mô đun này là tính điểm tổng hợp cho mỗi vận động viên.

Mỗi vận động viên sẽ tương ứng với một dãy các điểm số được lấy từ mảng **Diem**. Từ dãy này sẽ tính được điểm tổng hợp theo cách tính đặc biệt của BTC. Mô đun này sẽ cần hai hàm.

- Hàm **diem\_gk()** sẽ tính điểm tổng hợp các giám khảo từ dãy các điểm. Chú ý theo yêu cầu thì các giám khảo đầu tiên và cuối cùng có hệ số 2 nên cần tính điểm các giám khảo này 2 lần. Điều này được mô tả tại dòng lệnh 2. Hàm này được thiết kế chi tiết như sau:

---

```
1 def diem_gk(d):
2     diem = sum(d) + d[0] + d[len(d)-1]
3     diem = diem/(len(d)+2)
4     return round(diem,2)
```

---

- Hàm **Xuly()** sẽ tính toán điểm tổng hợp cho tất cả các vận động viên với đầu vào là dãy **Diem**. Hàm này sẽ sử dụng hàm **diem\_gk()** đã thiết lập ở phần trên. Kết quả tính toán là một dãy điểm tổng hợp và là giá trị trả lại của hàm này. Kết quả của hàm **Xuly()** là dãy **kq**.

---

```
1 def Xuly(B):
2     kq = []
3     for i in range(len(B)):
4         diem = diem_gk(B[i])
5         kq.append(diem)
6     return kq
```

---

### d) Thiết kế mô đun đưa kết quả ra

Cuối cùng là mô đun đưa kết quả ra tệp đầu ra theo yêu cầu của bài toán.

Hàm **ghiDL()** sẽ có ba tham số đầu vào là tên tệp dữ liệu ra, hai mảng **DS** và **kq**. Khai báo của hàm này là **ghiDL(fout, A, B)**, trong đó **fout** là tên tệp dữ liệu đầu ra, **A** là mảng danh sách các mã số của vận động viên, **B** là mảng ghi kết quả điểm tổng hợp của các vận động viên. Kết quả đưa ra màn hình.

---

```
1 def ghiDL(fout,A,B):
2     f = open(fout,"w")
3     for i in range(len(A)):
4         print(A[i],B[i],file = f)
5     f.close()
```

---

### e) Tổng hợp chương trình chính

---

```
1 finp = "SeaGames.inp"
2 fout = "ketqua.out"
3 DS,Diem = nhapDL(finp)
4 kq = Xuly(Diem)
5 ghiDL(fout,DS,kq)
```

---

Sau đây là toàn bộ chương trình:

### SeaGames.py

---

```
1 def nhapDL(finp):
2     f = open(finp)
3     A = []
4     B = []
5     for line in f:
6         S = line.split()
7         A.append(S[0])
8         temp = S[1:len(S)]
9         temp = [float(x) for x in temp]
10        B.append(temp)
11    f.close()
12    return A,B
13
14 def diem_gk(d):
15     diem = sum(d) + d[0] + d[len(d)-1]
16     diem = diem/(len(d)+2)
17     return round(diem,2)
18
19 def Xuly(B):
20     kq = []
21     for i in range(len(B)):
22         diem = diem_gk(B[i])
23         kq.append(diem)
24     return kq
25
26 def ghiDL(fout,A,B):
27     f = open(fout,"w")
28     for i in range(len(A)):
29         print(A[i],B[i],file = f)
30     f.close()
31
32 # Chương trình chính
33 finp = "SeaGames.inp"
34 fout = "ketqua.out"
35 DS,Diem = nhapDL(finp)
36 kq = Xuly(Diem)
37 ghiDL(fout,DS,kq)
```

---



## LUYỆN TẬP

1. Hãy chỉnh sửa lại chương trình trên nếu bổ sung thêm điều kiện sau vào nhiệm vụ: Trong tệp kết quả đầu ra, thứ tự các vận động viên được ghi theo thứ tự giảm dần của điểm đánh giá.
2. Trong nhiệm vụ trên, nếu công thức tính điểm tổng hợp của Sea Games thay đổi thì chúng ta có phải sửa lại toàn bộ chương trình hay không? Nếu cần thì chỉ phải sửa mô đun nào? Hàm nào?



## VẬN DỤNG

Thiết lập chương trình thiết kế theo mô đun cho các bài toán sau:

1. Cho trước số tự nhiên  $n$ , cần in ra trên màn hình dãy  $n$  số nguyên tố đầu tiên. Ví dụ nếu  $n = 5$  thì dãy cần in ra sẽ là 2, 3, 5, 7, 11.
2. Trong một kì thi Tin học trẻ, mỗi học sinh sẽ phải làm 3 bài thi. Với mỗi bài, nếu học sinh làm sẽ được ban giám khảo chấm và cho điểm, nếu không làm thì sẽ không tính điểm. Sau khi thi, dữ liệu điểm thi của học sinh sẽ được lưu trong một tệp văn bản và gửi về ban tổ chức. Mẫu một tệp điểm thi có dạng sau:

Diemthi.inp				
A12	12	-1	15	
B123	9	14	-1	
C11	10	12	18	
A110	10	-1	-1	
B01	12	10	4	

Quy định ghi trong tệp trên như sau:

- Mỗi dòng sẽ bắt đầu bằng số báo danh của thí sinh, tiếp theo là ba giá trị điểm tương ứng với ba bài thi.
- Điểm thi sẽ là một số tự nhiên từ 0 đến 20.
- Nếu học sinh không làm thi bài đó ghi -1.

Em có nhiệm vụ tính toán tổng số điểm thi của các bạn học sinh và đưa dữ liệu ra tệp **ketqua.out** là danh sách ba bạn có tổng điểm cao nhất được sắp xếp giảm dần từ trên xuống dưới. Khuôn dạng dữ liệu đưa ra bao gồm: số báo danh, các điểm thi từng bài và cuối cùng là tổng điểm cả ba bài.

Ví dụ với dữ liệu trên thì kết quả như sau:

ketqua.out				
C11	10	12	18	40
A12	12	-1	15	27
B01	12	10	4	26