

ปัญหา รายงานเกรด [GradeReport] (1 วินาที หน่วยความจำ 32 MB, 130 คะแนน)

ข้อมูลการลงทะเบียนของนักศึกษาในแต่ละเทอมประกอบไปด้วย

- (1) รหัสนักศึกษา เป็นเลขจำนวนเต็มบวก
- (2) จำนวนวิชาที่ลง
- (3) รายละเอียดแต่ละวิชาอันประกอบไปด้วย
 - (3.1) รหัสวิชาเป็นจำนวนเต็มบวก และ
 - (3.2) เกรดที่นักศึกษาได้เป็นจำนวนเต็ม

จะเห็นว่าข้อมูลเหล่านี้มีแค่เลขรหัส ไม่ได้มีชื่อนักศึกษาหรือชื่อวิชาอยู่เลย สาเหตุที่เป็นเช่นนั้นก็เพราะว่าระบบทะเบียนแยกเก็บชื่อนักศึกษากับรหัสไว้อีกชุดหนึ่ง และแยกเก็บรหัสวิชากับชื่อวิชาพร้อมทั้งจำนวนหน่วยกิตไว้อีกชุดหนึ่ง เมื่อเวลาจะพิมพ์รายงานออกมา ระบบทะเบียนจะใช้รหัสนักศึกษาไปค้นชื่อนักศึกษา เมื่อทราบชื่อนักศึกษาก็จะนำมาพิมพ์รวมกันในใบรายงาน สำหรับชื่อและจำนวนหน่วยกิตวิชาที่ดำเนินการในทำนองเดียวกัน

นอกจากนี้ยังมีเรื่องที่ต้องพิจารณาเพิ่มเติมอีกเล็กน้อย นั่นก็คือว่า ตัวเกรดนั้นเก็บเป็นเลขจำนวนเต็มเพื่อให้ง่ายต่อการคำนวณ แต่ตอนรายงานจริงระบบต้องการแสดงเกรดเป็นอักษรซึ่งมีอยู่ 6 แบบคือ A, B, C, D, F และ W ซึ่งตัวอักษรเหล่านี้มาจากเลข 4, 3, 2, 1, 0 และ -1

สุดท้ายแล้วระบบต้องคำนวณค่าคะแนนเกรด (Grade Point) รวมมาด้วย โดยคะแนนเกรดจากแต่ละวิชาจะคิดจาก (จำนวนหน่วยกิต * เกรดที่ได้) เช่น วิชาแคลคูลัสมี 4 หน่วยกิต หากได้เกรดเป็น B ก็จะมีคะแนนเกรดเป็น $4 \times 3 = 12$ คะแนน ส่วนวิชาคอมพิวเตอร์โปรแกรมมิ่งมี 3 หน่วยกิต หากได้เกรดเป็น C ก็จะมีคะแนนเกรดเป็น $3 \times 2 = 6$ คะแนน เป็นต้น อย่างไรก็ตามระบบจะไม่คิดคะแนนเกรดในวิชาที่ได้ W

จงเขียนโปรแกรมที่รับข้อมูลเกี่ยวกับการลงทะเบียน ข้อมูลชื่อนักศึกษา และ ข้อมูลชื่อวิชา จากนั้นพิมพ์สรุปรายงานเกรดนักศึกษาที่มีการลงทะเบียนออกมาเป็นผลลัพธ์

ข้อมูลเข้า

มีอยู่ 3 ส่วนด้วยกันดังนี้

ข้อมูลเข้าส่วนที่ 1 คือข้อมูลการลงทะเบียนของนักศึกษา ประกอบไปด้วย

บรรทัดแรกเป็นจำนวนเต็มบวก N โดยที่ $N \leq 100$ ซึ่งแทนจำนวนนักศึกษาที่มีการลงทะเบียนเรียน

ต่อมาจะมีข้อมูลอีก N ชุดเพื่อระบุข้อมูลการลงทะเบียนและเกรดของนักศึกษาที่มีการลงรายวิชา N คน

1. บรรทัดแรกจะเป็นรหัสนักศึกษา ซึ่งเป็นเลขจำนวนเต็มบวก ตามด้วย K ซึ่งแทนจำนวนวิชาที่ลง ($K \leq 10$)
2. ต่อมาอีก K บรรทัด จะเป็นเลขจำนวนเต็มสองตัวคือ รหัสวิชา และ เกรดที่นักศึกษาได้

ข้อมูลเข้าส่วนที่ 2 คือข้อมูลประจำตัวของนักศึกษา ประกอบไปด้วย

1. บรรทัดแรกเป็นจำนวนนักศึกษา P ในระบบทะเบียน (ซึ่งอาจจะมีมากกว่าจำนวนนักศึกษาที่ลงรายวิชา) และ $P \leq 1,000$
2. อีก P บรรทัดเป็นข้อมูลนักศึกษาสามรายการ แต่ละรายการคั่นด้วยช่องว่างหนึ่งช่อง ซึ่งมีลำดับดังนี้
 - 2.1. รหัสนักศึกษา เป็นเลขจำนวนเต็มบวก
 - 2.2. ชื่อแรกเป็นภาษาอังกฤษยาวไม่เกิน 20 ตัวอักษร ไม่มีช่องว่างใด ๆ
 - 2.3. นามสกุลเป็นภาษาอังกฤษยาวไม่เกิน 50 ตัวอักษร ไม่มีช่องว่างใด ๆ

ข้อมูลเข้าส่วนที่ 3 คือข้อมูลรายวิชา ประกอบไปด้วย

1. บรรทัดแรกคือ S แทนจำนวนรายวิชา โดยที่ $S \leq 1,000$
2. อีก S บรรทัดต่อมาเป็นข้อมูลแต่ละวิชา ซึ่งมีสามรายการคั่นด้วยช่องว่างหนึ่งช่อง ซึ่งมีลำดับดังนี้
 - 2.1. รหัสวิชา
 - 2.2. ชื่อวิชาเป็นภาษาอังกฤษ ยาวไม่เกิน 30 ตัวอักษร ไม่มีช่องว่าง
 - 2.3. จำนวนหน่วยกิตในวิชา เป็นเลขจำนวนเต็มบวกมีค่าไม่เกิน 6

หมายเหตุ รหัสนักศึกษาและวิชาจะไม่ขึ้นต้นด้วยเลขศูนย์แน่นอน และไม่ไม่เกิน 8 หลัก

ผลลัพธ์

เป็นรายงานเกรดนักศึกษา ซึ่งเรียงตามลำดับการปรากฏในข้อมูลเข้าส่วนที่ 1 และข้อมูลรายงานแต่ละคนจะเป็นดังนี้

1. บรรทัดแรก เป็นรหัสนักศึกษา ชื่อ และ นามสกุล ตามลำดับ
2. อีก K บรรทัดต่อมา (ค่า K แตกต่างกันได้สำหรับนักศึกษาแต่ละคน) เป็น รหัสวิชา ชื่อวิชา และเกรดแบบตัวอักษร ซึ่งลำดับของวิชา เรียงตามลำดับในข้อมูลเข้าส่วนที่ 1
3. ค่าผลรวมของ Grade Point

[ตัวอย่างอยู่ในหน้าถัดไป]

ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์
2	570123 Naruto Uzumaki
570123 3	111 ComPro W
111 -1	101 ComFound B
101 3	478 Calculus C
478 2	17
560234 2	560234 Itachi Uchiha
111 4	111 ComPro A
123 3	123 Golf B
3	18
560234 Itachi Uchiha	
570123 Naruto Uzumaki	
550234 Jonathan Jostar	
5	
111 ComPro 3	
101 ComFound 3	
123 Golf 2	
478 Calculus 4	
777 ResearchMethods 1	

คำแนะนำ มันอาจจะช่วยถ้าหากคุณจะมี struct หรือ class สามอันนี้ไว้ในโปรแกรมคุณ และคุณสร้างอาเรย์เก็บข้อมูลเหล่านี้แยกเป็นสามชุดไว้ก่อน จากนั้นจึงวนไปในอาเรย์ของเกรดตามลำดับ และค้นชื่อนักศึกษาและรายละเอียดวิชาออกมาก็จะได้โครงสร้างข้อมูลที่เชื่อมโยงกันได้โดยสะดวก หรือถ้าจะใช้ map เพราะเห็นว่า Key ต่าง ๆ มันเป็นจำนวนเต็มทำให้ใช้งานง่ายก็ได้เช่นกัน

```
struct grade {
```

```
} typedef Grade;
```

```
struct student {
```

```
} typedef Student;
```

```
struct subject {
```

```
} typedef Subject;
```