

- Tên học phần: **Ký thuật lập trình**
– Mã học phần: **TIN1053** – Số tín chỉ: **3** – Đề số: **1**
– Thời gian làm bài: **120 phút** (Không kể thời gian phát đề)
– Loại đề: **Không được sử dụng tài liệu**

Viết chương trình giải quyết những bài toán sau:

Câu 1. (3:0 điểm)

Thành phố MT lưu thông tin phạt người của mỗi xe ô tô bằng một chuỗi theo cú pháp:
[BS] [nx]: tròng dõ:

- BS là biển số xe (có 8 ký tự).
- n là một số nguyên ($0 \leq n \leq 9$) cho biết số lần phạm lỗi x.
- x là một chữ cái cho biết loại lỗi và mức phạt, được xác định theo bảng sau:

KÝ HIỆU LỖI	TÊN LỖI	MỨC PHẠT (TRIỆU ĐỒNG)	SỐ ĐIỂM GPLX BỊ TRỪ
4	Vượt đèn đỏ	18	4
l	Lấn làn	6	0
t	Ouá tốc độ	24	4
c	Nặng độ cón	24	6
Không phạm	Lỗi khác	1	0

Nếu một xe vi phạm nhiều loại lỗi thì các cặp [nx] được viết liên tiếp nhau. Ví dụ chuỗi cú pháp: 75A12345:2d3c4m có nghĩa là: Xe ô tô mang biển số 75A12345 vi phạm lỗi vượt đèn đỏ 2 lần, lấn làn 3 lần và lỗi khác 4 lần. Viết chương trình tính tiền phạt và số điểm GPLX bị trừ.

Đầu vào: Một chuỗi theo cú pháp như trên.

Đầu ra: Một chuỗi theo cú pháp: [BS]: [Số tiền phạt] [Số điểm GPLX bị trừ]. Nếu số điểm GPLX bị trừ lớn hơn hoặc bằng 12 thì ghi: "Tích thu GPLX".

Ví dụ:

INPUT	OUTPUT
75F53333:2l	75F3333:12 1
74F23522:2td.	74F2222:16 4
75A12455:2s4d3m	75A12453: 12 GPLX

chư số thập phân) và **kết quả**. Kết quả thi được xéi dựa trên điểm trung bình theo quy tắc

- Điểm trung bình dưới **6.00**: Không đạt giải.
- Điểm trung bình từ **6.00** đến dưới **7.50**: Giải khuyến khích.
- Điểm trung bình từ **7.50** đến dưới **8.50**: Giải bá.
- Điểm trung bình từ **8.50** đến dưới **9.50**: Giải nhì.
- Điểm trung bình từ **9.50** đến dưới **10**: Giải nhất.

Dầu vào:

- Dòng dầu chứa số nguyên n cho biết số thí sinh tham gia ($0 < n < 1000$).
- n dòng tiếp theo, mỗi dòng chứa các thông tin: **tên**, **điểm sàn phẩm** và **điểm trình bày**; cách nhau bởi một ký tự trống

Dầu ra: Gồm n dòng, mỗi dòng bao gồm **tên**, **điểm trung bình** và **kết quả**, cách nhau bởi một ký tự trống, sắp xếp theo thứ tự giảm dần của **điểm trung bình**. Nếu **điểm trung bình** bằng nhau thì sắp xếp theo thứ tự giảm dần của **điểm trình bày**.

Ví dụ:

INPUT	OUTPUT
3	Buoi 9.50 Giải nhất
Cam 8.8 5	Cam 8.23 Giải bá
Tao 5 5 5.5	Táo 5.50 Không đạt giải
Buoi 9.5 9.5	

Câu 3, (4.0 điểm)

Một danh sách liên kết dùng để quản lý nhân viên được tổ chức như sau:

```
typedef struct Employee {  
    int id;  
    string name;  
    int salary;  
    Employee *next;  
} Employee;
```

Hãy lập trình thực hiện các thao tác theo dữ liệu vào để quản lý danh sách nhân viên.

Dầu vào: Một dòng có một câu lệnh với các quy tắc như sau:

- **add <id> <tên> <luong>**: Bổ sung một nhân viên có mã, tên và lương vào cuối danh sách.
- **del <id>**: Xóa node có mã id id.
- **show <id>**: Tím kiếm và hiển thị thông tin của nhân viên có mã id theo định dạng: "ID x. Tên,x; Luong; x".
- **update <id> <money_noi>**: Cập nhật, lương mới của nhân viên có mã id.
- **show**: Hiển thị danh sách nhân viên, mỗi dòng có dạng: 'id - hame - salary'. Nếu danh sách rỗng thì hiển thị "DS RONG".
- **sort_salary**: Sắp xếp danh sách theo thứ tự giảm dần dựa vào lương.

- **count:** Hiển thị số lượng nhân viên trong danh sách.
- **avg:** Tính và hiển thị lương trung bình của tất cả nhân viên (làm tròn đến hai chữ số thập phân).
- **exit:** kết thúc chương trình.

Dấu ra: Tay vào cấu trúc lệnh nhận được để thực hiện và hiển thị kết quả.

Vì dụ:

INPUT	OUTPUT
add 101 hung 18000	101 - hung - 15000
add 102 long 12000	102 - long - 12000
add 103 hong 18000	103 - hong - 18000
show	ID. 102, Tên, long. Lương: 12000
find 102	3
count	15000.00
avg	101 - hung - 18000
sort_salary	102 - long—17000
show	105 - hong - 18000
update 102 20000	101 - hung - 15000
show	101 - hung - 15000
del 103	102 - long - 20000
show	101 - long - 20000)
count	2
avg	17500.00
exit	

Chú ý: Cần phải thực hiện việc cài đặt và thao tác trên danh sách liên kết đơn theo yêu cầu của đề bài.

(Ghi chú: Cần ký coi thì không được giải thích gì thêm)