

Quản lý thư viện

Giới hạn thời gian: 1.0s Giới hạn bộ nhớ: 256M

Một thư viện cần quản lý các lượt mượn sách trong ngày. Mỗi lượt mượn gồm các thông tin: mã lượt mượn, số ngày mượn, giá thuê mỗi ngày, và họ tên độc giả. Danh sách lượt mượn được quản lý bằng danh sách liên kết đơn với cấu trúc như sau:

```
struct Node {
    string maMuon;
    int soNgayMuon;
    int giaThue;
    string hoTen;
    Node* next;
};
Node* head = nullptr;
```

Hãy cài đặt và viết chương trình thực hiện các công việc như yêu cầu dưới đây.

Input

Đầu vào gồm nhiều dòng (tối đa 100 dòng), mỗi dòng là một trong các lệnh:

- `add_front <maMuon> <soNgayMuon> <giaThue> <hoTen>`: Thêm lượt mượn mới vào đầu danh sách. Mã lượt mượn phải duy nhất, nếu đã tồn tại thì in ra `"Da ton tai ma nay"`.
- `add_back <maMuon> <soNgayMuon> <giaThue> <hoTen>`: Thêm lượt mượn mới vào cuối danh sách. Mã lượt mượn phải duy nhất, nếu đã tồn tại thì in ra `"Da ton tai ma nay"`.
- `remove <maMuon>`: Xóa lượt mượn có mã `maMuon` khỏi danh sách, nếu không tồn tại thì in ra `"Khong tim thay"`.
- `print`: In toàn bộ danh sách lượt mượn từ đầu đến cuối, mỗi lượt trên một dòng theo định dạng:
`maMuon, soNgayMuon, giaThue, hoTen, thanhTien`
trong đó `thanhTien = soNgayMuon × giaThue`.
- `total`: In tổng doanh thu từ tất cả lượt mượn hiện có. Nếu danh sách rỗng, in `0`.
- `exit`: Kết thúc chương trình.

Output

- Với lệnh `print`: in danh sách lượt mượn theo định dạng đã cho.
- Với lệnh `total`: in tổng doanh thu.

Samples

Sample Input 1

```
add_front muon01 3 10000 Nguyen Van A
add_back muon02 5 8000 Le Thi B
add_front muon03 2 15000 Tran Van C
print
total
remove muon02
add_back muon01 1 5000 Nguyen Van D
print
total
remove muon04
exit
```

Sample Output 1

```
muon03, 2, 15000, Tran Van C, 30000
muon01, 3, 10000, Nguyen Van A, 30000
muon02, 5, 8000, Le Thi B, 40000
100000
Da ton tai ma nay
muon03, 2, 15000, Tran Van C, 30000
muon01, 3, 10000, Nguyen Van A, 30000
60000
Khong tim thay
```