#include <iostream>

#include <string>

using namespace std;

struct Expense {

string description;

float amount;

Expense\* next;

Expense(string desc, float amt) {

description = desc;

amount = amt;

next = nullptr;

}

};

struct Month {

string name;

Expense\* expenseHead;

Month\* next;

Month(string mName) {

name = mName;

expenseHead = nullptr;

next = nullptr;

}

};

class ExpenseTracker {

private:

Month\* head;

Month\* findMonth(string name) {

Month\* temp = head;

while (temp) {

if (temp->name == name) return temp;

temp = temp->next;

}

return nullptr;

}

public:

ExpenseTracker() {

head = nullptr;

}

void addMonth(string name) {

if (findMonth(name)) {

cout << "Month already exists.\n";

return;

}

Month\* newMonth = new Month(name);

newMonth->next = head;

head = newMonth;

cout << "Month added.\n";

}

void addExpense(string monthName, string desc, float amt) {

Month\* month = findMonth(monthName);

if (!month) {

cout << "Month not found.\n";

return;

}

Expense\* newExpense = new Expense(desc, amt);

newExpense->next = month->expenseHead;

month->expenseHead = newExpense;

cout << "Expense added.\n";

}

void updateExpense(string monthName, string oldDesc, string newDesc, float newAmt) {

Month\* month = findMonth(monthName);

if (!month) {

cout << "Month not found.\n";

return;

}

Expense\* exp = month->expenseHead;

while (exp) {

if (exp->description == oldDesc) {

exp->description = newDesc;

exp->amount = newAmt;

cout << "Expense updated.\n";

return;

}

exp = exp->next;

}

cout << "Expense not found.\n";

}

void deleteExpense(string monthName, string desc) {

Month\* month = findMonth(monthName);

if (!month) {

cout << "Month not found.\n";

return;

}

Expense\* curr = month->expenseHead;

Expense\* prev = nullptr;

while (curr) {

if (curr->description == desc) {

if (prev) prev->next = curr->next;

else month->expenseHead = curr->next;

delete curr;

cout << "Expense deleted.\n";

return;

}

prev = curr;

curr = curr->next;

}

cout << "Expense not found.\n";

}

void display() {

Month\* m = head;

while (m) {

cout << "Month: " << m->name << "\n";

Expense\* e = m->expenseHead;

while (e) {

cout << " - " << e->description << ": $" << e->amount << "\n";

e = e->next;

}

m = m->next;

}

}

void mostExpensive(string monthName) {

Month\* month = findMonth(monthName);

if (!month) {

cout << "Month not found.\n";

return;

}

Expense\* maxExp = nullptr;

Expense\* e = month->expenseHead;

while (e) {

if (!maxExp || e->amount > maxExp->amount)

maxExp = e;

e = e->next;

}

if (maxExp)

cout << "Most expensive: " << maxExp->description << " ($" << maxExp->amount << ")\n";

else

cout << "No expenses found.\n";

}

};

int main() {

ExpenseTracker tracker;

tracker.addMonth("January");

tracker.addExpense("January", "Rent", 1200);

tracker.addExpense("January", "Groceries", 250);

tracker.addExpense("January", "Utilities", 300);

tracker.display();

tracker.mostExpensive("January");

tracker.updateExpense("January", "Utilities", "Electricity", 400);

tracker.deleteExpense("January", "Groceries");

cout << "\nAfter updates:\n";

tracker.display();

tracker.mostExpensive("January");

return 0;

}