#include<string>
  
#include <iostream>
  
#include <string>
  
#include <vector>
  
#include <map>
  
#include <fstream>
  
  
using namespace std;
  
  
/\*
  
截止日期：10月30日
  
琳琳上大学后将她每天的消费支出项一行一项写在她的一个文本文件中，她的每一支出项目记录了支出日期、支出项目和金额。几个月过去了，她想统计一下她的消费行为信息，
  
请使用自定义单链表类，设计相应程序，帮助她完成相应的查询和统计。
  
  
具体功能可以自行设计，以下功能供参考：
  
（1）从文本文件读入所有n项支出项目；
  
（2）求出这n个支出项目中的最小、最大消费；每一天的平均消费；
  
（3）按照日期找出某一天的所有花费；
  
（4）按照项目找出该支出项目的所有花费。比如要求给出在“学习用品”这一项上总共花了多少钱；
  
（5）按照支出项花费递减的顺序输出每一项的对应总花费。
  
\*/
  
  
  
struct ShoppingNode
  
{
  
 string date;
  
 string project;
  
 float money;
  
 ShoppingNode \*next;
  
};
  
  
class LinkList
  
{
  
public:
  
 LinkList();
  
  
 LinkList(vector<ShoppingNode> v);
  
  
 LinkList(const string &filename);
  
  
 void AddShoppingNode(ShoppingNode \*n);
  
  
 void CoutMoney();
  
  
 void FindAllShoppingInOneDay(const string &date);
  
  
 int SumMoneyOfOneProject(const string &project);
  
  
 void DecrementOutput();
  
  
 vector<struct ShoppingNode> ReadFile(const string &filename);
  
  
 ShoppingNode \*GetFirstNode()
  
 { return this->first; };
  
  
 int GetLength()
  
 { return this->length; };
  
private:
  
 ShoppingNode \*first{};
  
 int length{};
  
};
  
  
LinkList::LinkList()
  
{
  
 first = new ShoppingNode;
  
 first->next = nullptr;
  
 this->length = 0;
  
}
  
  
LinkList::LinkList(vector<ShoppingNode> v)
  
{
  
 first = new ShoppingNode;
  
 auto r = first;
  
 ShoppingNode \*s = nullptr;
  
 this->length = v.size();
  
 for (int i = 0; i < this->length; i++)
  
 {
  
 s = new ShoppingNode;
  
 s->date = v[i].date;
  
 s->project = v[i].project;
  
 s->money = v[i].money;
  
 r->next = s;
  
 r = r->next;
  
 }
  
 r->next = nullptr;
  
}
  
  
void LinkList::CoutMoney()
  
{
  
 auto p = first->next;
  
 float sum\_money = 0;
  
 float average\_money = 0;
  
 float max\_money = -10000;
  
 float min\_money = 10000;
  
 string max\_date;
  
 string min\_date;
  
 string max\_project;
  
 string min\_project;
  
 while (p != nullptr)
  
 {
  
 sum\_money += p->money;
  
 if (p->money > max\_money)
  
 {
  
 max\_date = p->date;
  
 max\_project = p->project;
  
 max\_money = p->money;
  
 }
  
 if (p->money < min\_money)
  
 {
  
 min\_date = p->date;
  
 min\_project = p->project;
  
 max\_money = p->money;
  
 }
  
 p = p->next;
  
 }
  
 average\_money = sum\_money / this->length;
  
 cout << "最大的一笔消费发生在" << max\_date << ",购买的物品为" << max\_project << ",价格为" << max\_money << endl;
  
 cout << "最小的一笔消费发生在" << min\_date << ",购买的物品为" << min\_project << ",价格为" << min\_money << endl;
  
 cout << "总消费为" << sum\_money << ",平均消费为" << average\_money << endl;
  
}
  
  
void LinkList::FindAllShoppingInOneDay(const string &date)
  
{
  
 cout << "日期为" << date << "时的购买记录:" << endl;
  
 int count = 1;
  
 auto p = first->next;
  
 while (p != nullptr)
  
 {
  
 if (p->date == date)
  
 {
  
 cout << "第" << count << "笔消费为:" << endl;
  
 cout << "项目:" << p->project << endl;
  
 cout << "金额:" << p->money << endl;
  
 count++;
  
 }
  
 p = p->next;
  
 }
  
}
  
  
int LinkList::SumMoneyOfOneProject(const string &project)
  
{
  
 float sum = 0;
  
 auto p = first->next;
  
 while (p != nullptr)
  
 {
  
 if (p->project == project)
  
 {
  
 sum += p->money;
  
 }
  
 p = p->next;
  
 }
  
 cout << "项目" << project << "的总开销为" << sum << endl;
  
 return sum;
  
}
  
  
  
bool cmp(const pair<string, int> &a, const pair<string, int> &b)
  
{
  
 return a.second < b.second;
  
}
  
  
void LinkList::AddShoppingNode(ShoppingNode \*n)
  
{
  
  
}
  
  
void LinkList::DecrementOutput()
  
{
  
 map<string, float> project\_money;
  
 ShoppingNode \*p;
  
 p = first->next;
  
 while (p != nullptr)
  
 {
  
 if (project\_money.find(p->project) != project\_money.end())
  
 {
  
 project\_money[p->project] = project\_money[p->project] + p->money;
  
 } else
  
 {
  
 project\_money[p->project] = p->money;
  
 }
  
 p = p->next;
  
 }
  
 vector<pair<string, int> > vec(project\_money.begin(), project\_money.end());
  
 sort(vec.begin(), vec.end(), cmp);
  
 for (auto &it: vec)
  
 {
  
 cout << it.first << " " << it.second << endl;
  
 }
  
}
  
  
LinkList::LinkList(const string &filename)
  
{
  
 string temp;
  
 ShoppingNode sp;
  
 vector<ShoppingNode> v;
  
 ifstream file\_in(filename, ios::in);
  
 while (file\_in >> sp.date >> sp.project >> sp.money)
  
 {
  
 v.push\_back(sp);
  
 }
  
 first = new ShoppingNode;
  
 auto r = first;
  
 ShoppingNode \*s = nullptr;
  
 this->length = v.size();
  
 for (int i = 0; i < this->length; i++)
  
 {
  
 s = new ShoppingNode;
  
 s->date = v[i].date;
  
 s->project = v[i].project;
  
 s->money = v[i].money;
  
 r->next = s;
  
 r = r->next;
  
 }
  
 r->next = nullptr;
  
}
  
  
vector<ShoppingNode> LinkList::ReadFile(const string &filename)
  
{
  
  
 string temp;
  
 ShoppingNode sp;
  
 vector<ShoppingNode> v;
  
 ifstream file\_in(filename, ios::in);
  
 while (file\_in >> sp.date >> sp.project >> sp.money)
  
 {
  
 v.push\_back(sp);
  
 }
  
 return v;
  
}
  
  
int main()
  
{
  
 LinkList(a){"ShoppingData.txt"};
  
  
 a.CoutMoney();
  
 a.FindAllShoppingInOneDay("2022.9.1");
  
 a.DecrementOutput();
  
  
  
 return 0;
  
}