## 1070 Milk

### 一、题目

#### 问题描述

Here are some rules:

1. Ignatius will never drink the milk which is produced 6 days ago or earlier. That means if the milk is produced 2005-1-1, Ignatius will never drink this bottle after 2005-1-6(inclusive).

#### 输入数据

The input contains several test cases. The first line of the input is a single integer T which is the number of test cases. T test cases follow. Each test case starts with a single integer N(1<=N<=100) which is the number of kinds of milk. Then N lines follow, each line contains a string S(the length will at most 100 characters) which indicate the brand of milk, then two integers for the brand: P(Yuan) which is the price of a bottle, V(mL) which is the volume of a bottle.

#### 输出数据

For each test case, you should output the brand of the milk which is the cheapest. If there are more than one cheapest brand, you should output the one which has the largest volume.

#### 输入样例

2 2 Yili 10 500 Mengniu 20 1000 4 Yili 10 500 Mengniu 20 1000 Guangming 1 199 Yanpai 40 10000

#### 输出样例

Mengniu Mengniu

*Hint*

In the first case, milk Yili can be drunk for 2 days, it costs 10 Yuan. Milk Mengniu can be drunk for 5 days, it costs 20 Yuan. So Mengniu is the cheapest.In the second case, milk Guangming should be ignored. Milk Yanpai can be drunk for 5 days, but it costs 40 Yuan. So Mengniu is the cheapest.

#### 题目来源

HDU 1070 http://acm.hdu.edu.cn/showproblem.php?pid=1070

### 二、题解

#### 解题思路

题目大意为挑选出最经济的喝牛奶方式。首先要确认的是，每盒奶只喝五天。 小于200ml的奶直接忽略，之后我们可以通过计算每天的平均花费而不是每毫升的平均花费。 比如一盒价格30元的999ml的奶只能喝4天，用999整除200，得到4。然后(float)30/4 = 7.25元每天。 如果一盒价格30元的9999ml的奶，那肯定只能喝5天了。然后(float)30/5 = 6元每天。 倘若两种奶的日花销相同，那么就只挑量大的那盒买

#### 参考程序

#include <stdio.h>  
#include <string.h>  
#include <math.h>  
  
char str[110][110];  
char strMax[110];  
int P[110];  
int V[110];  
float DC[110];  
  
float cheapest;  
int cheapestV;  
  
int main()  
{  
 int C;  
 int N;  
 scanf("%d", &C);  
 while(C--)  
 {  
 scanf("%d", &N);  
 int count = 0;  
 while(N--)  
 {  
 scanf("%s%d%d", str[count], &P[count], &V[count]);  
 if(V[count] < 200)  
 continue;  
 else  
 {   
 if(V[count] >= 1000)  
 DC[count] = (float)P[count] / 5;  
 else  
 DC[count] = (float)P[count] / (float)(V[count] / 200);  
 count++;  
 }  
 }  
 cheapest = DC[0];  
 cheapestV = V[0];  
 strcpy(strMax, str[0]);  
 for(int j = 1; j < count; j++)  
 {  
 if(cheapest > DC[j])  
 {  
 cheapest = DC[j];  
 cheapestV = V[j];  
 strcpy(strMax, str[j]);  
 }  
 else if(cheapest == DC[j])  
 {  
 if(V[j] > cheapestV)  
 {  
 cheapest = DC[j];  
 cheapestV = V[j];  
 strcpy(strMax, str[j]);  
 }  
 }  
 }  
 printf("%sn", strMax);   
 }  
}

#### 复杂度分析

O(n)

#### 编程技巧

无