

温州大学计算机与人工智能学院

程序设计基础(2023-2024-1) 课程作业

实验名称	谁拿了最多奖学金				
班 级	23大数据1	姓 名	徐王晶	学 号	23211870102
实验地点	南5B105	实验时间	2023-10-14,22:24:27	指导老师	虞铭财

一、问题编号：

0212

地址：<http://10.132.254.54/problem/212/>

二、问题描述：

某校的惯例是在每学期的期末考试之后发放奖学金。发放的奖学金共有五种，获取的条件各自不同：

- 1) 院士奖学金，每人8000元，期末平均成绩高于80分（>80），并且在本学期内发表1篇或1篇以上论文的学生均可获得；
- 2) 五四奖学金，每人4000元，期末平均成绩高于85分（>85），并且班级评议成绩高于80分（>80）的学生均可获得；
- 3) 成绩优秀奖，每人2000元，期末平均成绩高于90分（>90）的学生均可获得；
- 4) 西部奖学金，每人1000元，期末平均成绩高于85分（>85）的西部省份学生均可获得；
- 5) 班级贡献奖，每人850元，班级评议成绩高于80分（>80）的学生干部均可获得； 只要符合条件就可以得奖，每项奖学金的获奖人数没有限制，每名同学也可以同时获得多项奖学金。

例如姚林的期末平均成绩是87分，班级评议成绩82分，同时他还是一位学生干部，那么他可以同时获得五四奖学金和班级贡献奖，奖金总数是4850元。 现在给出若干学生的相关数据，请计算哪些同学获得的奖金总数最高（假设总有同学能满足获得奖学金的条件）。

三、输入说明：

输入的第一行是一个整数N（ $1 \leq N \leq 100$ ），表示学生的总数。

接下来的N行每行是一位学生的数据，从左向右依次是姓名，期末平均成绩，班级评议成绩，是否是学生干部，是否是西部省份学生，以及发表的论文数。

姓名是由大小写英文字母组成的长度不超过20的字符串（不含空格）；期末平均成绩和班级评议成绩都是0到100之间的整数（包括0和 100）；是否是学生干部和是否是西部省份学生分别用一个字符表示，Y表示是，N表示不是；发表的论文数是0到10的整数（包括0和10）。每两个相邻数据项之间用一个空格分隔。

四、输出说明：

输出包括三行，第一行是获得最多奖金的学生的姓名，第二行是这名学生获得的奖金总数。如果有两位或两位以上的学生获得的奖金最多， 输出他们之中在输入文件中出现最早的学生的姓名。第三行是这N个学生获得的奖学金的总数。

五、输入样例：

```
4
YaoLin 87 82 Y N 0
ChenRuiyi 88 78 N Y 1
LiXin 92 88 N N 0
ZhangQin 83 87 Y N 1
```

六、输出样例：

```
ChenRuiyi
9000
28700
```

七、解答内容：

所用语言：

源代码：

```
01. #include <stdio.h>
02.
03. int main(void)
04. {
05.     char name[100][50];
06.     int avr_score[100];
07.     int score[100];
08.     char gb[100][4];
09.     char xb[100][4];
10.     int lw[100];
11.     int money[100] = {0};
12.     int p = 0;
13.
14.     int n;
15.     scanf("%d", &n);
16.
17.     while (n--)
18.     {
19.         scanf(
20.             "%s %d %d %s %s %d",
21.             name[p],
22.             &avr_score[p],
23.             &score[p],
24.             gb[p],
25.             xb[p],
26.             &lw[p]
27.         );
28.         p++;
29.     }
30.
31.     int index_max_money = 0;
32.     int max_money = 0;
33.     int sum_money = 0;
34.
35.     for (int i = 0; i < p; i++)
36.     {
37.         if (avr_score[i] > 80 && lw[i] >= 1)
38.         {
39.             money[i] += 8000;
40.         }
41.         if (avr_score[i] > 85 && score[i] > 80)
42.         {
43.             money[i] += 4000;
44.         }
45.         if (avr_score[i] > 90)
46.         {
47.             money[i] += 2000;
48.         }
49.         if (avr_score[i] > 85 && xb[i][0] == 'Y')
50.         {
51.             money[i] += 1000;
52.         }
53.         if (score[i] > 80 && gb[i][0] == 'Y')
54.         {
55.             money[i] += 850;
56.         }
57.
58.         sum_money += money[i];
59.         if (money[i] > max_money)
60.         {
61.             max_money = money[i];
62.             index_max_money = i;
63.         }
64.     }
65.
66.     printf(
67.         "%s\n%d\n%d\n",
68.         name[index_max_money],
69.         money[index_max_money],
70.         sum_money
71.     );
72.
73.     return 0;
74. }
```

八、判题结果

AC - 答案正确

判题结果补充说明：

test id: 470,result:AC, usedtime:0MS, usedmem:812KB,score:10 test id: 605,result:AC, usedtime:0MS, usedmem:812KB,score:10
test id:2667,result:AC, usedtime:0MS, usedmem:812KB,score:10 test id:2668,result:AC, usedtime:0MS,
usedmem:812KB,score:10 test id:2669,result:AC, usedtime:0MS, usedmem:812KB,score:10 test id:2670,result:AC,
usedtime:0MS, usedmem:812KB,score:10 test id:2671,result:AC, usedtime:0MS, usedmem:812KB,score:10 test
id:2672,result:AC, usedtime:0MS, usedmem:812KB,score:10 test id:2673,result:AC, usedtime:0MS, usedmem:812KB,score:10
test id:2674,result:AC, usedtime:0MS, usedmem:812KB,score:10