

泥塑课

小米是一个幼儿园老师，每学期的泥塑课上，她都会给每个学生发不超过250立方厘米的等量橡皮泥，教大家做泥塑。在上课过程中，她发现每个班都恰好有一个小朋友会去抢另一个小朋友的橡皮泥，于是她决定，在正式开始做泥塑前，让大家把手里的橡皮泥都捏成一个立方体，并且测量手里捏好的橡皮泥的长、宽和高。这样，她就可以知道谁被谁抢了橡皮泥了。小米老师在不同的学期可能会带一个班或者同时带多个班，因此输入数据可能有一组或者多组。每组输入数据的第一行为一个整数n，表示了这个班的小朋友数，之后n行每行包括了由空格分隔的三个整数和一个字符串，那个字符串表示了小朋友的名字，前面三个整数则是这个学术手里橡皮泥块的长、宽、高数据。按照幼儿园的规定，每个班最多有9个小朋友，最少也要有2个小朋友，每个小朋友在学籍系统中的名称不超过8个字符长。当出现一个班级的小朋友数为-1时，表示没有更多的班级了。输出行数与小米老师带的班级数相同，形式为“X took clay from Y.”，具体请参考样例输出。

```
3
10 10 2 Jill
5 3 10 Will
5 5 10 Bill
4
2 4 10 Cam
4 3 7 Sam
8 11 1 Graham
6 2 7 Pam
-1
```

输出：

```
Bill took clay from Will.
Graham took clay from Cam.
```

下面是我编的，哪里出错了？

```
#include<stdio.h>
struct x
{
    int a, b, c;
    char name[8];
};
int main()
{
    struct x x[8];
    int i, j, max=0, min=0, n, s[8];    //还有没打完
    while(scanf("%d", &n)!=-1)
    {
        for(i=0; i<n; i++)
        {
            scanf("%d %d %d %s", &x[i].a, &x[i].b, &x[i].c, &x[i].name);
```

```

    }
    for(i=0; i<n; i++)
    {
        s[i]=x[i].a*x[i].b*x[i].c;
    }
    for(i=0; i<n-1; i++)          //求谁抢了橡皮泥
    {
        for(j=i+1; j<n; j++)
        {
            if(s[i]<s[j])
            {
                max=j;
                i=j;
            }
            else
                max=i;
        }
    }
    for(i=0; i<n-1; i++)          //求谁橡皮泥被抢了
    {
        for(j=i+1; j<n; j++)
        {
            if(s[i]>s[j])
            {
                min=j;
                i=j;
            }
            else
                min=i;
        }
    }          //知道谁抢谁了
    printf("%s took clay from %s.\n", x[max].name, x[min].name);
}
return 0;
}

```

而我的输出是：

Bill took clay from Will.

Graham took clay from Pam.

Graham took clay from Pam.

此处存疑，明天再去思考

2.官方答案

C 实现

```

1  #include <stdio.h>
2  #define NUM_OF_CLASSROOM 20 //设定课室数组上限
3  int main(){
4  //初始化课室数量，内容为每个课室的人数，amountClass表示课室的数量
5  int numOfStudent[NUM_OF_CLASSROOM] = { 0 }, amountClass = 0;
6
7  //记录学生姓名指针的二维数组，
8  //首位为课室号，次位为学生学号（由0开始计），末位表示最长为8个字符。
9  //调用例：&name[2][3]可得到班别为2，学号为3的同学的姓名
10 char name[NUM_OF_CLASSROOM][9][9];
11
12 int V[NUM_OF_CLASSROOM][9]; //记录每个学生所拥有泥塑的体积
13 inti, j;
14 int x, y, z; //长x宽y高z，用以计算体积
15 int max, min; //某课室中拥有泥塑体积最大max及最小min的学号
16
17 //输入数据
18 scanf("%d", &numOfStudent[0]);
19 while (numOfStudent[amountClass] != -1){
20 for (i = 0; i < numOfStudent[amountClass]; i++){
21 scanf("%d %d %d %s", &x, &y, &z, &name[amountClass][i]);
22 //算出该班所有学生所拥有的泥塑体积
23 V[amountClass][i] = x * y * z;
24 }
25 scanf("%d", &numOfStudent[++amountClass]);
26 }
27
28 //找出每间课室拥有泥塑体积最大和最小的学生，并输出
29 for (i = 0; i < amountClass; i++){
30 min = max = 0;
31 for (j = 0; j < numOfStudent[i]; j++){
32 if (V[i][max] < V[i][j])
33 max = j;
34 if (V[i][min] > V[i][j])
35 min = j;
36 }
37 printf("%s took clay from %s.", &name[i][max], &name[i][min]);
38 if (i != amountClass - 1)
39 printf("\n");
40 }
41
42 return 0;
43 }

```

它如何工作

输入部分

输入部分使用while语句和for语句嵌套，while语句和for语句虽然都是循环，但在应用场景上略有不同，while语句更适用于循环次数不确定的情况，而for语句适合循环次数确定的情况。

在输入的同时，应该马上算出该班该学生的体积，以免再来一个循环去计算，因为泥塑的长宽高是不重要的，体积才是我们要的东西。

在while语句里面的scanf()函数里，我使用了 ++amountClass 这种称为“前置自增”的方式，与 amountClass++ 这样的后置自增方式不同，前置自增是在语句执行前首先自增，再去执行所在的语句，而后置自增的方式则是在语句执行完毕后再自增的。在当前的情况下，适宜使用前置自增。

处理并输出部分

首先用for语句遍历所有课室，然后查看该课室每个学生所拥有的泥塑体积，找出拥有泥塑体积最大和最小的同学（因为其他同学的体积都是相等的），找出最大最小后，按照指定的格式输出对应的名字则可。

3.

```
#include "stdio.h"
```

```

struct ban{
    int chang;
    int kuan;
    int gao;
    char name[9];

```

```

    int sum;//总体积
}stu[9];

void main()
{
    int i=0;//循环用
    int x=0;//学生数
    int min,max;
    //int sum;//总体积

    int Student[9]={0};
    scanf("%d", &Student[x]);
    while(Student[x]!=-1)
    {
        min=0;
        max=0;
        for(i=0;i<Student[x];i++)
        {
            scanf("%d %d %d %s",&stu[i].chang,&stu[i].kuan,&stu[i].gao,&stu[i].name);
            stu[i].sum=stu[i].chang*stu[i].kuan*stu[i].gao;
            // printf("%d\n",stu[i].sum);
        }

        for(i=0;i<Student[x];i++)
        {
            if(stu[i].sum>stu[max].sum)
            {
                max=i;
            }
            if(stu[i].sum<stu[min].sum)
            {
                min=i;
            }
        }

        printf("%s took clay from %s.\n",&stu[max].name,&stu[min].name);
        //scanf("%d",&x);
        scanf("%d", &Student[++x]);
    }
}

```

```
}
```

```
4 .
```

```
#include <stdio.h>
```

```
int main () {
```

```
int n = 0;
```

```
while (1) {
```

```
scanf("%d",&n);
```

```
if (n == -1) break;
```

```
char str[10][10];
```

```
int min = 1000, max = 0, mini = 0, maxi = 0;
```

```
for (int i = 0; i < n; i++) {
```

```
int a, b, c, x;
```

```
scanf("%d %d %d %s",&a,&b,&c,&str[i]);
```

```
x = a * b * c;
```

```
if (x < min) {
```

```
min = x;
```

```
mini = i;
```

```
}
```

```
if (x > max) {
```

```
max = x;
```

```
maxi = i;
```

```
}
```

```
}
```

```
printf("%s took clay from %s.\n",str[maxi],str[mini]);
```

```
}
```

```
return 0;
```

```
}
```