中国人民大学
RENMIN UNIVERSITY OF CHINA

第8讲 结构体

余力

buaayuli@ruc.edu.cn

从“#195 平均成绩排序”说起

有 n 位学生，每位学生修读的科目数不尽相同，已知所有学生的各科成绩，要求按学生平均成绩由高到低输出学生的学号、平均成绩；当平均成绩同时，按学号从低到高排序。对平均成绩，只取小数点后前 2 位，从第 3 位开始舍弃（无需舍入）。

输入格式

□□输入为 n+1 行，第一行为 n 表示学生人数。

□□从第二行开始的 n 行，每行为一名学生的成绩信息，包括：学号、科目数，各科成绩。其中 n、学号、成绩均为整数，它们的值域为：0≤n≤10000，1≤学号≤1000000，0≤成绩≤100。学生的科目数都不超过 100 门。

输出格式

□□最多 n 行，每行两个数，学号在前，后为平均成绩，空格分隔。若 n 为 0，输出 NO；若某学生所修科目不到 2 门，则不纳入排序，若无人修满 2 门，也输出 NO。

输入样例	输出样例
5	2003 93.75
1001 2 89 78	1001 83.50
2003 4 88 99 100 88	1004 72.66
4004 3 72 80 66	4004 72.66
1004 3 70 66 82	
3001 1 100	

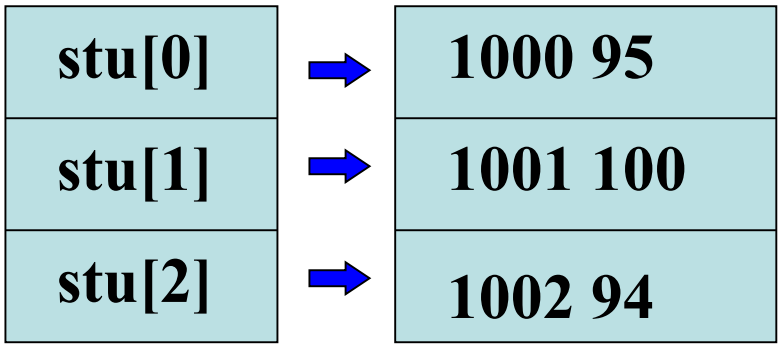
```
int main() {
    int id[10000];
    double aver[10000], tmpf;
    int n, i, j, StuNo, KemuNum, tmp, sum, count = 0;

    scanf ("%d", &n);
    // 输入 n 位学生信息
    for (i = 0; i < n; i++) {
        scanf ("%d %d", &StuNo, &KemuNum);
        for (sum = 0, j = 0; j < KemuNum; j++) {
            scanf ("%d", &tmp);
            sum = sum + tmp; }
        if (KemuNum >= 2) {
            id[count] = StuNo;
            aver[count] = sum * 1.0 / KemuNum;
            count++; }
    }
```

```
if (count == 0) {
    printf("NO");
    return 0; }
else {
    for (i = 0; i < count - 1; i++)
        for (j = 0; j < count – i-1; j++)
            if ((fabs(aver[j] - aver[j + 1]) < 1e-7 && id[j] > id[j + 1]) || aver[j] < aver[j + 1]) {
                tmp = id[j]; id[j] = id[j + 1]; id[j + 1] = tmp;
                tmpf = aver[j]; aver[j] = aver[j + 1]; aver[j + 1] = tmpf; }
    for (i = 0; i < count; i++)
        printf( "%d %.2lf\n", id[i], int(aver[i] * 100) / 100.0 );
    return 0;
    }
```

核心问题

- 应该如何表示一名学生的信息?
  - 表示为二维数组，例：stu[1000][2]
  - 每一行表示一名学生
  - 第一列表示学号
  - 第二列表示平均成绩

```
样例一
5
1000 95
1001 100
1002 94
1006 95
1007 100
```



中国人民大学
RENMIN UNIVERSITY OF CHINA

1. 为什么结构体

为什么要使用结构体 (1)

- 考虑下面的实际问题
  - 将一组学生按照出生日期进行排序
- 待排序的“学生”不是基本数据类型!
- 学生信息包括
  - 姓名：汉语拼音，最多20个字符
  - 性别：M / F
  - 生日：19841107（年月日）
  - 身高：1.74（m）
  - 体重：51.5（kg）

为什么要使用结构体 (2)

- 使用现有学的知识完成此任务？
  - 定义5个数组，分别对应多项信息
  - 在排序算法中同时操作这5个数组

#308 猪场分配

老赵经营着一家现代化畜牧养殖企业，在全国各地建有 n 家专业养猪场。但是由于受非洲猪瘟的影响，2019 年损失较为惨重。在国家政策和市场需求的激励下，老赵决定分三个批次购入仔猪，恢复生产。为了杜绝疫病交叉感染的潜在风险，同一个批次的只能放在同一个养殖场。由于不同场地的养殖成本与容量不同，现在他需要考虑如何安排养殖场，以实现最佳的经济效益。假定各批次出栏时，市场预期价格一致，根据输入的养殖场现状信息，编程找出一种总成本最少的猪场分配方案（不是成本最少的所有方案，见输出说明）。

【输入格式】

第 1 行 3 个整数，分别表示三个批次的仔猪数量。

第 2 行 1 个整数，表示老赵经营的养殖场数 n。

第 3 行开始，共 n 行，每行 5 个整数，依次表示：养殖场的编号、运营状态、最大养殖容量、运营基础成本和每头仔猪的出栏养殖成本。其中运营状态用 0、1 表示，1 表示已在运营，不能安排其它批次仔猪进场。比如：112 1 3000 5000 300。

【输出格式】

如果找到最少成本的方案，输出两行，第 1 行只 1 个数，为最少成本；第 2 行 3 个数，为对应该成本的分配方案，即依仔猪批次顺序，输出养殖场的编号。这些编号是在所有可能最佳方案中，各批次可能分配的猪场最小编号。

如果找不到，输出 NO。

```
int Farm_Total, Farm_Count, P1_Num, P2_Num, P3_Num, P1_id, P2_id, P3_id;
int ID[3001], rongliang[3001], base_cost[3001], each_cost[3001], data[3001][5];
```

【数据说明】

1）所有数字的输入输出均空格分隔；

2）总成本在整数的有效范围内，不会超过 int 型的最大值；

3）60%的测试数据 n ≤ 100；100% 的测试数据 n ≤ 3000。

02. 结构体定义与使用

结构体类型的定义

- 结构体：表示一组类型可能不同的数据

```
struct Student {
    char  name[20];              //姓名
    char sex;                    //性别
    unsigned  long birthday; //生日
    float height;                //身高
    float weight;                //体重
};
```

```
struct farm {
        int num;
        int state;
        int max_cap;
        int basic_cost;
        int pig_cost;
};
```

```
struct Stu {
        int id;
        int kemu;
        double aver;
};
```

定义结构体类型的格式

```
struct 结构体名
{
    类型名 1   成员名 1;
    类型名 2   成员名 2;
      . . .
    类型名 n   成员名 n;
};
```

struct 是结构体类型的标志，结构体名student是编程者自己选定

编程习惯：首字母大写！

大括号所括起来的5条语句是结构体中5个成员的定义。

结构体定义之后一定要跟一个“；”号。

结构体变量的定义与引用 (1)

```
#include <stdio.h>

struct  Student                    //定义结构
{
    char  name[20];                //姓名
    char sex;                     //性别
    unsigned  long birthday;    //生日
    float height;                  //身高
    float weight;                  //体重
 };
```

结构体变量的定义与引用 (2)

```
int main()      {                                   // 主函数
  struct Student my;  // 结构体变量printf(  “输入自己的数据
    \n”  ); // 提示信息
    printf( "姓名（汉语拼音）\n" );  // 显示提示
    printf( "性别：M/F\n" );
    printf( "生日（年月日）\n" );
    printf( "身高（米）\n" );
    printf( "体重（kg）\n\n" );
```

结构体变量的定义与引用 (3)

```
// 依次输入个人信息
  scanf( "%s", my.name);
  scanf( "%c", &my.sex );
  scanf( "%d", &my.birthday );
  scanf( "%f", &my.height );
  scanf( "%f", &my.weight );
```

结构体变量的定义与引用 (4)

```
    // 依次输出个人信息
    printf( "%s\n", my.name );
    printf( "%c\n", my.sex );
    printf( "%d\n", my.birthday );
    printf( "%f\n", my.height );
    printf( "%f\n", my.weight );
    return 0;
}
```

说明

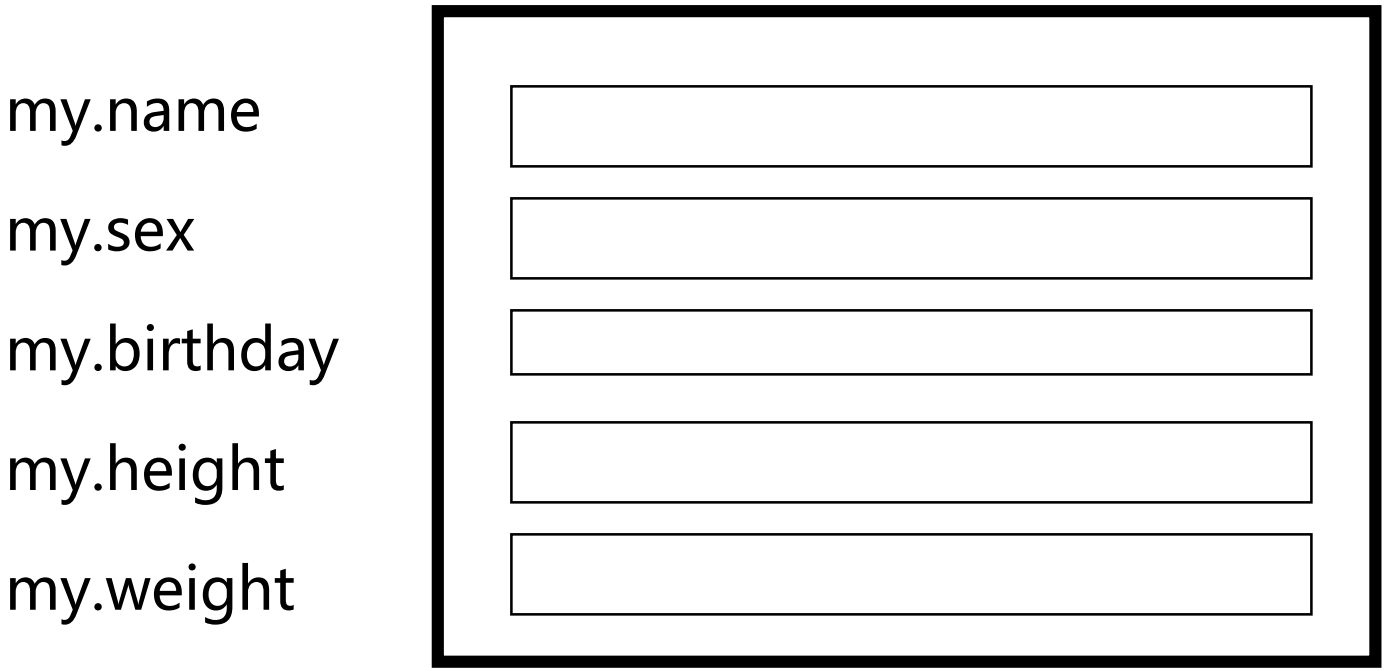
- 定义类型不会分配内存空间

```
struct Student     //此处为结构类型定义
 {
     char  name[20];                  //姓名
     char sex;                        //性别
     unsigned  long birthday;         //生日
     float height;                    //身高
     float weight;                    //体重
 };
```

//结构变量my的定义，系统会为my分配内存空间

```
struct Student my;
```

变量 my （my为变量的符号地址）



& my (变量的内存地址)

结构体变量的初始化

- 方法一：定义和初始化同时完成

```
struct  Person
{
    char  name[10];
    unsigned  long  birthday;
    char  placeofbirth[20];
} per = { “Li ming”, 19821209, “Beijing”};
```

Person 是结构体类型
per 是结构体变量
per.name = “Liming”;
per.birthday = 19821209;
per.placeofbirth = “Beijing”;

结构体变量的初始化

- 方法二：分开完成

```
struct  Person
{
    char  name[10];
    unsigned  long  birthday;
    char  piaceofbirth[20];
 };

struct Person per = { “Li ming” , 19821209,  “Beijing” };
```

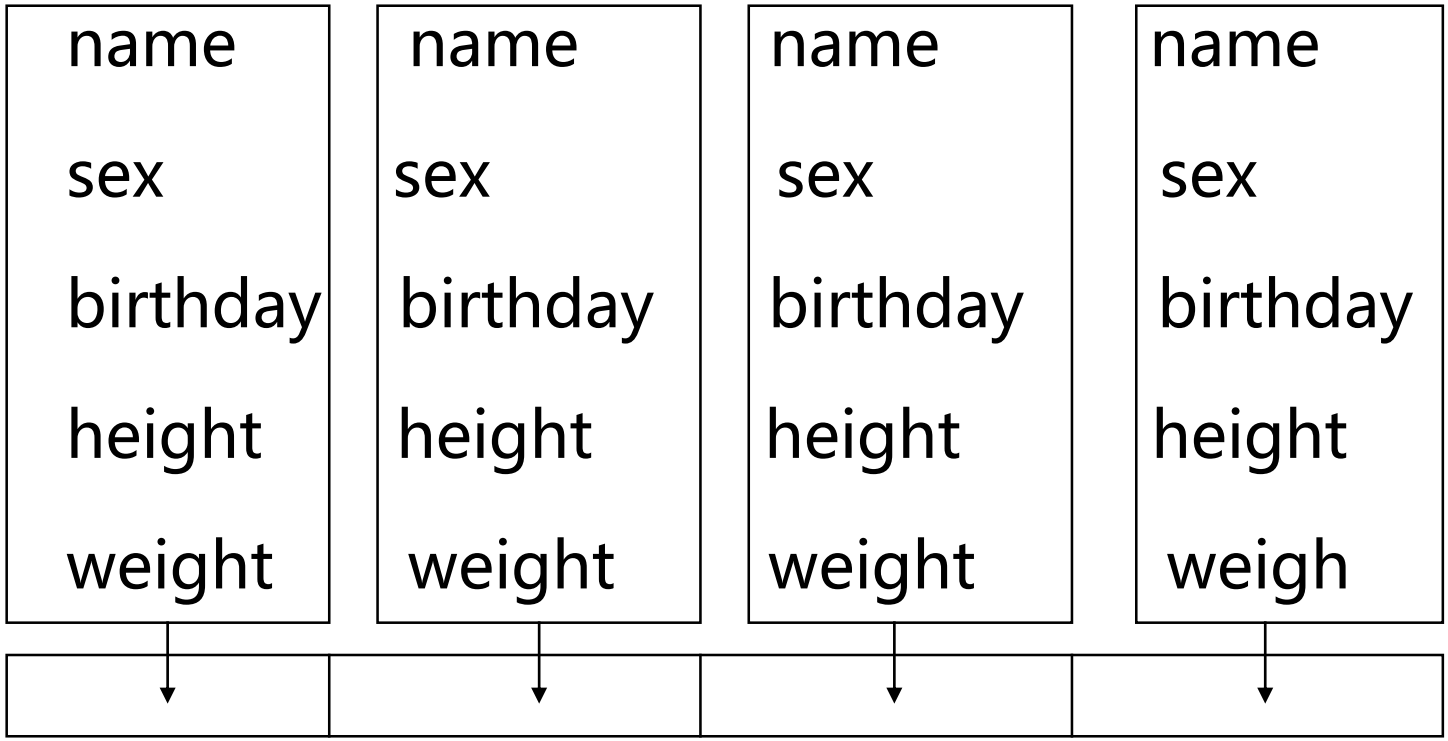
中国人民大学
RENMIN UNIVERSITY OF CHINA

03. 结构体数组

结构数组

- 结构也可以构成数组，即每个数组元素是一个结构；
- 当然，要求这一类数组的全部元素都应该是同一类结构；
- 例如：**同宿舍4名同学的数据，构成一个有4个**元素的结构数组。

```
struct Student  Room [ 4 ] = {
        { “Li li”,  ‘M’, 19840318, 1.82, 65.0 },
        { “Mi mi”,  ‘M’, 19830918, 1.75, 58.0 },
        { “He lei”,  ‘M’, 19841209, 1.83, 67.1 },
        { “Ge li”,  ‘M’, 19840101, 1.70, 59.0 }
};
```


name
sex
birthday
height
weight
name
sex
birthday
height
weight
name
sex
birthday
height
weight
name
sex
birthday
height
weigh
下标0
下标1
下标2
下标3
Room

Room[0]	“Li li”, ‘M’, 19840318, 1.82, 65.0
Room[1]	“Mi mi”, ‘M’, 19830918,1.75, 58.0
Room[2]	“He lei”, ‘M’, 19841209, 1.83, 67.1
Room[3]	“Ge li”, ‘M’, 19840101, 1.70, 59.0

```
#include <stdio.h>
struct  Student  //名为student的结构类型
{
    char  name[20];              //姓名
    char sex;                    //性别
    unsigned  long birthday;   //生日
    float height;                //身高
    float weight;                //体重
 };
struct Student  Room [ 4 ] = { //定义全局结构数组/初始化
      {“Li li”, ‘M’, 19840318, 1.82, 65.0 },
      {“Mi mi”, ‘M’, 19830918, 1.75, 58.0 },
      {“He lei”, ‘M’, 19841209, 1.83, 67.1 },
      {“Ge li”, ‘M’, 19840101, 1.70, 59.0 }
};
```

```
int  main() {
    struct Student q;      //结构变量q，用于交换时保存中间结果
    int i = 0 ; int j =0;
   //按照年龄排序
    for ( j = 0; j < 3; j = j+1 )
       for ( i = 0; i < 3 – j ; i = i +1 )
            if ( Room[ i ].birthday >  Room[i+1].birthday )
                 {    q= Room[ i ];   //交换，结构变量赋值
                      Room[ i ]=Room[ i+1];
                      Room[ i+1]= q;
                 }

   for( i = 0; i < 4; i = i + 1 ) {    //输出排序后的结果
            printf( "%s\n", Room[i].name );
            printf( "%c\n", Room[i].sex );
            printf( "%d\n", Room[i].birthday );
            printf( "%f\n", Room[i].height );
            printf( "%f\n", Room[i].weight );
     }
    return 0;
}
```

#195 平均成绩排序（结构体）

```
struct Stu
{
    int id;
    int kemu;
    double aver;
};

int main(){
    int n,i,j,k,sum,lin1,lin2,z=0,x,lo;
    struct Stu stu[10000];
    struct Stu tmp;
    scanf("%d",&n);
```

```
int main(){
    int n,i,j,k,sum,lin1,lin2,z=0,x,lo;
    struct Stu stu[10000];
    struct Stu tmp;
    scanf("%d",&n);
```

```
    // 输入 n 位学生信息
    for(i=0;i<n;i++)
        {sum=0;
        scanf("%d %d",&lin1,&lin2);
        if(lin2<2)
            for(int u=0;u<lin2;u++)
                scanf("%d",&lin1);
        else //输入学生的多门课成绩并求好平均数
            {stu[z].id=lin1; stu[z].kemu=lin2;
            for(j=0;j<stu[z].kemu;j++)
                {scanf("%d",&x);
                sum=sum+x;
                }
            stu[z].aver=sum*1.0/stu[z].kemu;
            z++; //重新计算保存有 2 门以上课程的学生
            }
    }
```

#195 平均成绩排序

```
    if(z==0){
        printf("NO");
        return 0;
    }
    else{
    
        // 先按学号排序（第 2 排序依据）
        for(i=0;i<z-1;i++){
            k=0;
            for(j=0;j<z-1-i;j++)
                if(stu[j].id>stu[j+1].id)
                    {tmp=stu[j];stu[j]=stu[j+1];stu[j+1]=tmp;  k=1;  }
            if(k==0)break;// 表明已经排序好了
            }
        // 再按平均成绩排序（第 1 排序依据）
        for(i=0;i<z-1;i++){
            k=0;
            for(j=0;j<z-1-i;j++)
                if(stu[j].aver<stu[j+1].aver)   {  tmp=stu[j];   stu[j]=stu[j+1];  stu[j+1]=tmp;  k=1;}
            if(k==0)  break;
        }
```

#308 猪场分配

老赵经营着一家现代化畜牧养殖企业，在全国各地建有 n 家专业养猪场。但是由于受非洲猪瘟的影响，2019 年损失较为惨重。在国家政策和市场需求的激励下，老赵决定分三个批次购入仔猪，恢复生产。为了杜绝疫病交叉感染的潜在风险，同一个批次的只能放在同一个养殖场。由于不同场地的养殖成本与容量不同，现在他需要考虑如何安排养殖场，以实现最佳的经济效益。假定各批次出栏时，市场预期价格一致，根据输入的养殖场现状信息，编程找出一种总成本最少的猪场分配方案（不是成本最少的所有方案，见输出说明）。

【输入格式】

第 1 行 3 个整数，分别表示三个批次的仔猪数量。

第 2 行 1 个整数，表示老赵经营的养殖场数 n。

第 3 行开始，共 n 行，每行 5 个整数，依次表示：养殖场的编号、运营状态、最大养殖容量、运营基础成本和每头仔猪的出栏养殖成本。其中运营状态用 0、1 表示，1 表示已在运营，不能安排其它批次仔猪进场。比如：112 1 3000 5000 300。

【输出格式】

如果找到最少成本的方案，输出两行，第 1 行只 1 个数，为最少成本；第 2 行 3 个数，为对应该成本的分配方案，即依仔猪批次顺序，输出养殖场的编号。这些编号是在所有可能最佳方案中，各批次可能分配的猪场最小编号。

如果找不到，输出 NO。

```
int Farm_Total, Farm_Count, P1_Num, P2_Num, P3_Num, P1_id, P2_id, P3_id;
int ID[3001], rongliang[3001], base_cost[3001], each_cost[3001], data[3001][5];
```

【数据说明】

1）所有数字的输入输出均空格分隔；

2）总成本在整数的有效范围内，不会超过 int 型的最大值；

3）60%的测试数据 n ≤ 100；100% 的测试数据 n ≤ 3000。

#308 猪场分配

```
#include <stdio.h>
int main() {
    int Farm_Total, Farm_Count, P1_Num, P2_Num, P3_Num, P1_id, P2_id, P3_id;
    int ID[3001], rongliang[3001], base_cost[3001], each_cost[3001], data[3001][5];
    int i, j, k, find;
    int cost_sum, min_cost = 10000000;

    scanf("%d%d%d%d", &P1_Num, &P2_Num, &P3_Num, &Farm_Total);

    for (i = 0, Farm_Count = 0; i < Farm_Total; i++) {
        for (j = 0; j < 5; j++)
            scanf("%d", &data[i][j]);
        if (data[i][1] == -1) continue;
        ID[Farm_Count] = data[i][0];
        rongliang[Farm_Count] = data[i][2];
        base_cost[Farm_Count] = data[i][3];
        each_cost[Farm_Count] = data[i][4];
        Farm_Count++;//记录可用场子数
    }
```

```
for (i = 0; i < Farm_Count; i++) {
    if (rongliang[i] < P1_Num)  continue;
    for (j = 0; j < Farm_Count; j++) {
        if (rongliang[j] < P2_Num || j == i)  continue;
        for (k = 0; k < Farm_Count; k++) {
            if (rongliang[k] < P3_Num || k == j || k == i)  continue;
            cost_sum = base_cost[i] + base_cost[j] + base_cost[k];
            cost_sum = cost_sum + each_cost[i] * P1_Num + each_cost[j] * P2_Num + each_cost[k] * P3_Num;
            find = 0;
            if (cost_sum < min_cost)  find = 1;
            else if (cost_sum == min_cost)
                if (ID[i] < P1_id) find = 1;
                else if (ID[i] == P1_id)
                    if (ID[j] < P2_id) find = 1;
                    else if (ID[j] == P2_id && ID[k] < P3_id)  find = 1;
            if (find == 1) {
                min_cost = cost_sum;
                P1_id = ID[i]; P2_id = ID[j]; P3_id = ID[k]; }
        }
    }
}
```

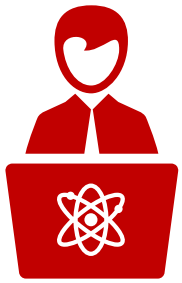
#308 猪场分配

```
#include <stdio.h>
int main() {
    int s1, s2, s3, n, i, j, k, a, b, c, temp;
    int sum = 0, summin = 1000000000;
    struct pig {
        int b;
        int t;
        int m;
        int basic;
        int per;
    } fac[3000] = {0};
    scanf("%d %d %d", &s1, &s2, &s3);
    scanf("%d", &n);
    for (i = 0; i < n; i++)
        scanf("%d %d %d %d %d", &fac[i].b, &fac[i].t, &fac[i].m, &fac[i].basic, &fac[i].per);
```

#308 猪场分配

```
for (i = 0; i < n; i++)
    if ((s1 <= fac[i].m) && fac[i].t == 0)
        for (j = 0; j < n; j++)
            if (j == i)     continue;
            else if ((s2 <= fac[j].m) && (fac[j].t == 0))
                for (k = 0; k < n; k++)
                    if ((k == j) || (k == i))  continue;
                    else if ((s3 <= fac[k].m) && (fac[k].t == 0)) {
                        sum = fac[i].basic + fac[j].basic + fac[k].basic + s1 * fac[i].per + s2 * fac[j].per + s3 * fac[k].per;
                        if (sum < summin) {
                            summin = sum;
                            a = fac[i].b;
                            b = fac[j].b;
                            c = fac[k].b;
                        }
                    }
```

中國人民大學
RENMIN UNIVERSITY OF CHINA



谢谢大家！