

# 结构体

## 结构体的概念

在 C 语言中，结构体（struct）是一种自定义的数据类型，用于把不同类型的数据组合在一起，形成一个整体。

### ★ For Example

学生的信息包括：学号（int）、姓名（char数组）、成绩（float）

用结构体，就可以把学号、姓名、成绩组合成一个整体，像一个“学生”对象

### ★ 核心作用：

可以把相关数据组织在一起，便于管理。

提高程序的可读性和可维护性。

在数组中可以存放多个结构体，实现复杂数据集合。

## 结构体的定义

```
struct 结构体名 {  
    数据类型1 成员名1;  
    数据类型2 成员名2;  
    ...  
};
```

```
struct Student {  
    int id;          // 学号  
    char name[50];   // 姓名  
    float score;    // 成绩  
};
```

定义结构体只是定义了类型，还没创建变量。

结构体变量可以在定义后单独声明，或者在定义时顺便声明。

## 结构体变量的声明

### 两种方式

#### 方法一：

先定义结构体类型，再声明变量

#### 方法二：

定义结构体的同时声明变量

```
struct Student s1; // 声明一个 Student 类型的变量 s1
```

```
struct Student {  
    int id;  
    char name[50];  
    float score;  
} s1, s2; // 声明了两个 Student 类型的变量
```

# 访问结构体成员

```
s1.id = 101;  
strcpy(s1.name, "张三");  
s1.score = 95.5;
```

如果是结构体指针，使用 -> 运算符：

```
struct Student *p = &s1;  
p->score = 100;
```

## 结构体数组

```
struct Student students[3]; // 定义 3 个学生的数组  
students[0].id = 101;  
strcpy(students[0].name, "张三");  
students[0].score = 90;
```

### 例题



定义一个学生结构体，包含学号、姓名、成绩，并输入 3 个学生信息，最后输出学生成绩大于 80 的信息。

#### 1-定义结构体

```
struct Student {  
    int id;  
    char name[50];  
    float score;  
};
```

#### 2-定义结构体数组

```
struct Student students[3];
```

#### 3. 输入学生信息

```
for(int i = 0; i < 3; i++) {  
    scanf("%d %s %f", &students[i].id, students[i].name, &students[i].score);  
}
```

#### 4. 输出成绩大于 80 的学生

```
for(int i = 0; i < 3; i++) {  
    if(students[i].score > 80) {  
        printf("学号: %d, 姓名: %s, 成绩: %.2f\n",  
               students[i].id, students[i].name, students[i].score);  
    }  
}
```

## 题目描述

[复制 Markdown](#) [展开](#) [进入 IDE 模式](#)

现有  $N$  名同学参加了期末考试，并且获得了每名同学的信息：语文、数学、英语成绩（均为不超过 150 的自然数）。如果某对学生成绩的分差都不大于 5，且总分分差不大于 10，那么这对学生就是“旗鼓相当的对手”。现在想知道这些同学中，有几对“旗鼓相当的对手”？同样一个人可能会和其他好几名同学结对。

## 输入格式

第一行一个正整数  $N$ 。

接下来  $N$  行，每行三个整数，其中第  $i$  行表示第  $i$  名同学的语文、数学、英语成绩。最先读入的同学编号为 1。

## 输出格式

输出一个整数，表示“旗鼓相当的对手”的对数。

## 输入输出样例

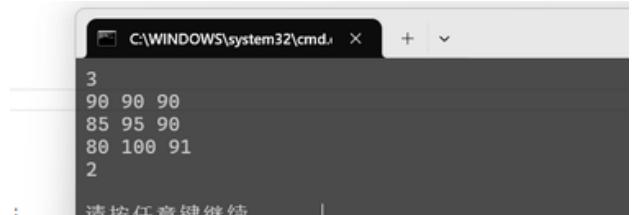
输入 #1	输出 #1
3 90 90 90 85 95 90 80 100 91	2

## 说明/提示

数据保证， $2 \leq N \leq 1000$  且每科成绩为不超过 150 的自然数。

- 已知  $N$  名学生，每名学生有 语文、数学、英语三科成绩。
- 定义“旗鼓相当的对手”：成绩差

## 统计有多少学生对符合条件



```

1 #include <stdio.h>
2 #include <stdlib.h>
3 // 1. 定义结构体
4 struct Student {
5     int chinese;
6     int math;
7     int english;
8 };
9 int main() {
10     int N;
11     scanf("%d", &N);
12     struct Student students[N];
13     // 2. 输入学生成绩
14     for (int i = 0; i < N; i++) {
15         scanf("%d %d %d", &students[i].chinese, &students[i].math, &students[i].english);
16     }
17     int count = 0;
18     // 3. 两两比较学生
19     for (int i = 0; i < N; i++) {
20         for (int j = i + 1; j < N; j++) {
21             int total_i = students[i].chinese + students[i].math + students[i].english;
22             int total_j = students[j].chinese + students[j].math + students[j].english;
23             if (abs(students[i].chinese - students[j].chinese) <= 5 && abs(students[i].math - students[j].math) <= 5 &&
24                 abs(students[i].english - students[j].english) <= 5 &&
25                 abs(total_i - total_j) <= 10) {
26                 count++;
27             }
28         }
29     }
30     // 4. 输出结果
31     printf("%d\n", count);
32     return 0;
33 }
```

# 结合了排序算法

【样例输入】

3

amethystic 12345.67  
amethyst 6543.21  
wangwei 7645.434

【样例输出】

amethyst 6543.210  
wangwei 7645.434  
amethystic 12345.670

## 【问题描述】

编写一个程序，输入 N 个员工的姓名和工资，按照工资从小到大顺序输出员工的姓名和工资。

## 【输入形式】

用户首先在第一行输入一个正整数，该正整数表示待排序的员工数目，然后在下面多行输入多个员工的信息，每行的输入格式为：姓名 工资。以回车结束每个员工的输入。

## 【输出形式】

程序输出排序后的结果。每行的输出结果格式也是：姓名 工资。姓名和工资字段中间用空格分隔，工资为浮点数类型，输出保留小数点后面三位小数，假定员工姓名不能超过 20 个字符，用户的数量不超过 50 个。

```
#include <stdio.h>
#include <string.h>
struct Employee {
    char name[21];
    float salary;
};
int main() {
    int N;
    scanf("%d", &N);
    struct Employee emp[N];
    // 1. 输入员工信息
    for (int i = 0; i < N; i++) {
        scanf("%s %f", emp[i].name, &emp[i].salary);
    }
    // 2. 冒泡排序按工资升序
    for (int i = 0; i < N - 1; i++) {
        for (int j = 0; j < N - 1 - i; j++) {
            if (emp[j].salary > emp[j + 1].salary) {
                struct Employee temp = emp[j];
                emp[j] = emp[j + 1];
                emp[j + 1] = temp;
            }
        }
    }
    // 3. 输出结果
    for (int i = 0; i < N; i++) {
        printf("%s %.3f\n", emp[i].name, emp[i].salary);
    }
    return 0;
}
```

**注意：**  
**浮点数输出格式**  
**排序算法的选择**

```
C:\WINDOWS\system32\cmd.exe
3
main()
int N;
scanf("%d", &N);
struct Employee emp[N];
// 1. 输入员工信息
for (int i = 0; i < N; i++) {
    scanf("%s %f", emp[i].name, &emp[i].salary);
}
// 2. 冒泡排序按工资升序
for (int i = 0; i < N - 1; i++) {
    for (int j = 0; j < N - 1 - i; j++) {
        if (emp[j].salary > emp[j + 1].salary) {
            struct Employee temp = emp[j];
            emp[j] = emp[j + 1];
            emp[j + 1] = temp;
        }
    }
}
// 3. 输出结果
for (int i = 0; i < N; i++) {
    printf("%s %.3f\n", emp[i].name, emp[i].salary);
}
return 0;
```

## 6 图书目录排序

【样例输入】

3

Cprogramming 48.00

Java 32.00

C++ 50.00

【样例输出】

Java 32.00

Cprogramming 48.00

C++ 50.00

### 【问题描述】

编写一个程序，输入 N 本书的书名和价格，按照图书价格从小到大顺序排列输出图书的书名和价格。

### 【输入形式】

用户首先在第一行输入一个正整数，该正整数表示待排序的图书数目，然后在下面多行输入多个图书的信息，每行的输入格式为：书名 价格。以回车结束每本图书的输入。

### 【输出形式】

程序输出排序后的结果。每行的输出结果格式也是：书名 价格。书名和价格字段中间用空格分隔，假定书名不超过 20 个字符，价格为浮点数类型，输出保留小数点后面两位小数，图书的数量不超过 50 个。

```
1 #include <stdio.h>
2 #include <string.h>
3 struct Book {
4     char name[21];
5     float price;
6 };
7 int main() {
8     int N;
9     scanf("%d", &N);
10    struct Book books[N];
11    // 1. 输入图书信息
12    for (int i = 0; i < N; i++) {
13        scanf("%s %f", books[i].name, &books[i].price);
14    }
15    // 2. 冒泡排序按价格升序
16    for (int i = 0; i < N - 1; i++) {
17        for (int j = 0; j < N - 1 - i; j++) {
18            if (books[j].price > books[j + 1].price) {
19                struct Book temp = books[j];
20                books[j] = books[j + 1];
21                books[j + 1] = temp;
22            }
23        }
24    }
25    for (int i = 0; i < N; i++) {
26        printf("%s %.2f\n", books[i].name, books[i].price);
27    }
28    return 0;
29 }
```

```
C:\WINDOWS\system32\cmd.exe + ^
```

```
3
Cprogramming 48.00
Java 32.00
C++ 50.00
Java 32.00
Cprogramming 48.00
C++ 50.00
```