# 实验十  结构体与单链表

## 【目的与要求】

1．掌握结构体类型的说明、结构体变量(数组)的定义及初始化方法；

2．掌握结构体变量成员的引用；。

## 【上机内容】

### **【一般示例】**

#### 结构体类型声明与变量定义

结构定义的一般形式为：

struct 结构体名{ 数据类型1 成员名1；

…

数据类型n 成员名n；

};

其中，关键字struct用来引入一个结构说明，或指出一个结构定义的开始。“结构体名”是对该结构体类型的命名。一对大括号中定义了组成该结构的成员项，每个成员项由成员数据类型、成员名称和一个分号组成，一个结构中成员项的多少、顺序没有限制。

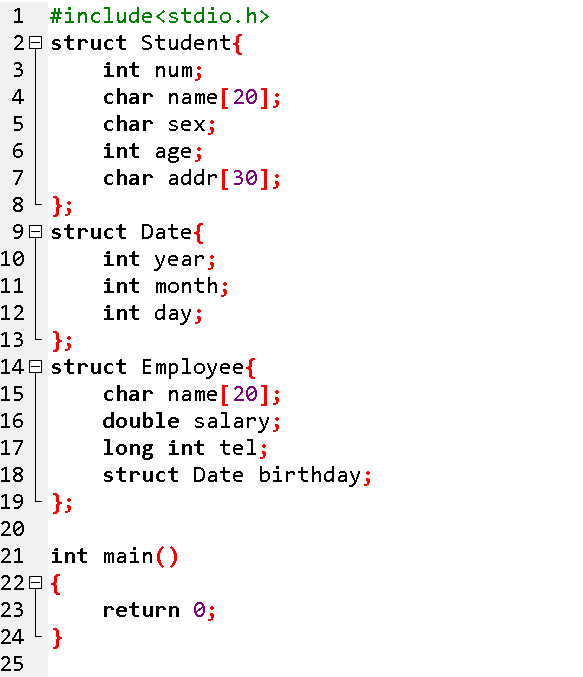
“数据类型i”用来指出对就成员的数据类型，它可以是C语言的数据类型名，如int、float、char、double等等；成员类型也可以为一个结构体类型。

“成员名i”就是组成该结构体类型的成员i的名字，它按标识符的组成规则确定，习惯上将结构体名第一个字母大写。

结构定义是一个整体，在程序中作为一个定义语句出现。包括诸成员项说明的一对大括号和最后的分号不能遗漏，否则引起错误。因为它是一个定义语句，所以一个结构定义可以在函数的内部和函数的外部进行。

最后必须强调，关于上述形式的结构体定义仅仅是描述了一个数据结构的“模型”，或者说只是定义了一种名为“结构体名”的数据类型，C编译程序并不给这样的定义分配对就的空间，因此它不是一个实在的数据对象。

例：



可以使用以上三种方法定义结构体类型变量：

**（一）先声明结构体类型，再定义结构体变量名**

一旦一个结构类型被定义，便可以像用C固有的数据类型说明符定义变量那样来定义这种结构类型的变量，其定义形式是：

struct 结构体名 结构变量名1，结构变量名2，…，结构变量名n；

其中“结构体名”是已经定义过的结构体类型名。如Date,Employee等。“结构体变量名i”即是要定义的与指定结构体名相关的变量名，它们都按标识符的构成规则确定。

如：

struct Date birthday, nationalday, hiredate;

从上面的例子可以得到如下结论：

类型与变量是不同的概念。只有在定义了结构体类型名之后才能定义结构体变量，否则C编译程序无法给结构变量分配存储空间；

结构全变量的存储空间分配是按照其对应的结构体类型的诸成员项的定义顺序进行；

同一结构类型的每个结构变量的存储空间大小都是相同的，且是各成员项所占空间之和。这个空间可经sizeof运算符对该结构体名进行运算得到；

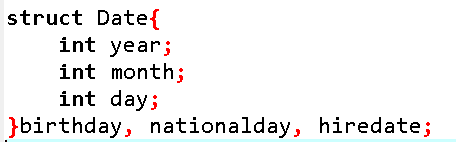
如执行语句：

 运行结果是

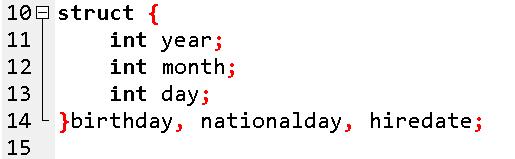


成员也可以是一个结构体变量。

**（二）在声明结构体类型的同时定义变量**



**(三)直接定义结构类型变量**



上面定义结构体变量birthday, nationalday, hiredate;时省略了结构体类型名称。这种结构体定义是“一次性的”，不能在其他程序段再定义该结构体变量。例如在主函数中再使用语句：



会导致编译器报错。

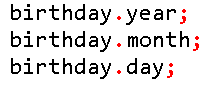
#### 结构体引用

结构变量的一般表示形式为：

结构变量名．成员名;

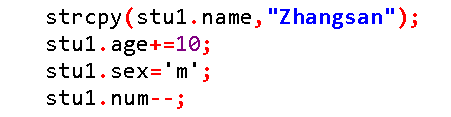
这样的组合形式既是结构变量成员的表示形式，也是它的访问形式。其中的“.”号称之为“结构成员运算符”，它用来连接结构体名与成员名，具有“从属于”的含义，即其后的成员名是前面的结构变量中的一个成员。

如上面示例中的结构变量birthday，它的三个成员分别表示为：



结构体成员具有与普通变量完全相同的性质，可以像普通变量那样参于各种运算。既可以出现在赋值号的左边向它赋值，也可以出现在赋值号的右边作为一个运算分量参于表达式的计算，可以作为“＋＋”、“－－”、“&”等之类的运算符的操作数。

下面是一些正确的结构变量成员的表示、访问的例子。



注意：

（1）不能将一个结构体变量作为一个整体进行输入和输出。如**不能**作如下引用：

printf(“%d, %s, %c, %d, %f, %s\n”, stu1);

（2）对于嵌套结构体变量的访问：

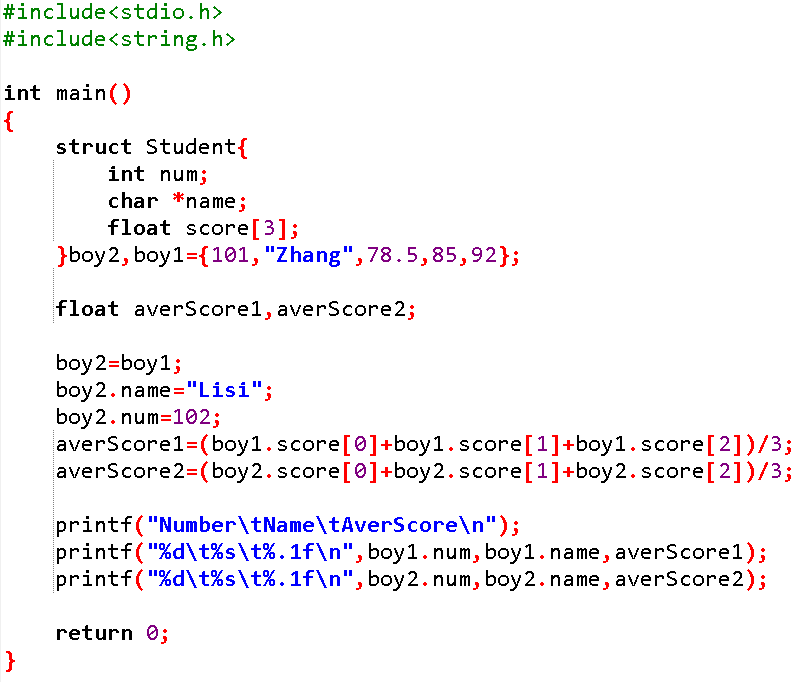


（3）可以引用结构体变量成员的地址，也可以引用结构体变量的地址。如：



#### 结构体程序范例

下列程序演示了使用结构体表示学生信息初始化，赋值，输入的一般过程。





### **【实验任务】**

下表是某个小学班级学籍信息，使用结构体有关知识，完成下列至少3个任务：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 学号 | 语文 | 数学 | 英语 |
| Xiaoming | 2001 | 100 | 98 | 85.5 |
| Hangmeimei | 2002 | 95.5 | 98 | 85.5 |
| …… |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

#### 学生成绩录入与输出

设计结构体并声明变量，存储学生学号，姓名，语，数，外三门课程成绩。

函数

DataInput()

DataOutput()

分别完成数据输与输出

【输入要求】

N+1行：

第1行，输入一个正整数N（N <= 100），表示学生人数。

第2至接着输入N+1行，第1~N个学生信息，每行格式如下：

学号 姓名 分数1 分数2 分数3

分数是一个非负整数，且小于等于100；

姓名为一个连续的字符串，中间没有空格，长度不超过20。

【输出要求】

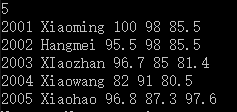
N+1行：

第1行，表头，第2~N+1行，学生成绩信息。

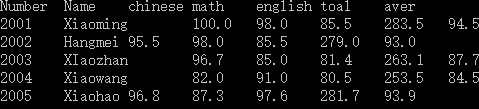
成绩精确到小数点后面一位

学号 姓名 分数之间的间距为一个Tab

【输入样例】



【输出样例】



#### 2.学生成绩排名

在上例基础上，编写StuSort()函数，实现学生总分排名，并按成绩从高至低输出学生信息。

【输入要求】

N+1行：

第1行，输入一个正整数N（N <= 100），表示学生人数。

第2至接着输入N+1行，第1~N个学生信息，每行格式如下：

学号 姓名 分数1 分数2 分数3

分数是一个非负整数，且小于等于100；

姓名为一个连续的字符串，中间没有空格，长度不超过20。

【输出要求】

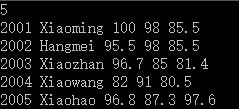
N+1行：

第1行，表头，第2~N+1行，排序后的学生成绩信息。

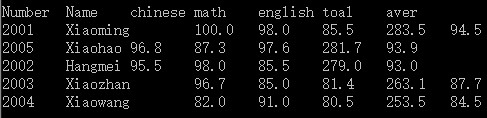
成绩精确到小数点后面一位

学号 姓名 分数之间的间距为一个Tab

【输入样例】



【输出样例】



#### 3. 学生排名查找

设计结构体，存储学生学号，姓名，语，数，外三门课程成绩后，进行学生总分排名，编写int searchStu(struct student stu[],int n,char stuname[])函数，根据姓名查找学生排名的查找，并输出所查找到学生信息，不允许重名。

【输入要求】

N+2行

第1行，输入一个正整数N（N <= 100），表示学生人数。

第2至接着输入N+1行，第1~N个学生信息

每行格式如下：

学号 姓名 分数1 分数2 分数3

分数是一个非负整数，且小于等于100；

姓名为一个连续的字符串，中间没有空格，长度不超过20。

第N+2行，软入所要查找的学生姓名；

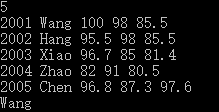
【输出要求】

1行或2行：

如果找到所输入姓名的学生信息，第1行输出名次，第2行输出学生信息。

如果没找到，输出“Not found!”

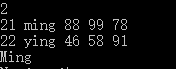
【输入样例】



【输出样例】



【输入样例】



【输出样例】



#### 学生信息链表

设计结构体并声明变量，存储学生学号，姓名，语，数，外三门课程成绩。

函数原型说明：

stuct student \* DataInput()，使用库函数mollocc动态生成学生链表节点，当输入学号为0或负数时，终止数据输入，并返回链表头节点。

DataOutput(stuct student \*head)，打印输出链表。

【输入要求】

若干行：

输入格式：依次输入“学号 姓名 成绩1 成绩2 成绩3”，使用空格或回车分隔数据输入。如输入学号为0或负数，则终止输入。

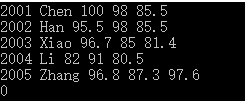
【输出要求】

若干行：

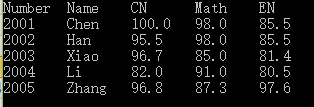
第1行为表头；

第2行以后为学生信息，每行1个学生，数据使用转义符“\t”分开，小数精确到1位。

【输入样例】



【输出样例】



【输入样例】



【输出样例】



#### 学生信息链表之查找最高分

设计结构体并声明变量，存储学生学号，姓名，语，数，外三门课程成绩。

函数原型说明：

stuct student \* DataInput()，使用库函数mollocc动态生成学生链表节点，当输入学号为0或负数时，终止数据输入，并返回链表头节点。

struct student \*searchmax(struct student \*head)，查找总分最高的学生信息，并返回该学生在链表中地地址，主函数打印输出查到的学生信息。

【输入要求】

若干行：

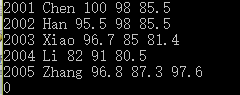
输入格式：依次输入“学号 姓名 成绩1 成绩2 成绩3”，使用空格或回车分隔数据输入。如输入学号为0或负数，则终止输入。

【输出要求】

0行或1行：

如找到符合条件的学生，则输出该学生信息，每行1个学生，数据使用转义符“\t”分开，小数精确到1位。

【输入样例】



【输出样例】



【输入样例】



【输出样例】

无输出

单链表实验

#### 1建立简单的静态链表

10101 89.5

10102 90.0

10103 85.0

建立一个如图所示的简单链表，将上述3个学生数据以链表形式表示，每个结点包括学号、姓名和下一个结点等成员，打印输出链表中各结点的数据。

【输入要求】

无

【输出要求】

3行：

第一行：学号\t成绩

第二行：学号\t成绩

第三行：学号\t成绩

【输入样例】

无

【输出样例】

10101 89.5

10102 90.0

10103 85.0

#### 2 单链表反转

输入一组整数，以链表形式存储，编写void Reserve (ListNode \*h)，将链表反转。

【输入要求】

1行：

第一行：输入数字的个数

第二行：输入n个数字

【输出要求】

1行：

反转后的数字

【输入样例】

6

1 2 3 4 5 6

【输出样例】

6 5 4 3 2 1

#### 3 约瑟夫（链表）

用链表实现约瑟夫问题：13个人围成一圈，从第1个人开始顺序报号1，2，3。凡报到3者退出圈子。找出最后留在圈子中的人原来的序号。

【输入要求】

无

【输出要求】

2行：

第1行，出圈的人的序号；

第2行：The last one is+最后留在圈子中的人的原来序号

提示：格式控制为%4d

【输入样例】

无

【输出样例】

3 6 9 12 2 7 11 4 10 5 1 8

The last one is 13

#### 4 删除结点

使用结构体存储学生信息，包括学号、姓名、性别（m/f）、年龄，保存在链表中。

输入年龄，删除链表中该年龄的节点。

【输入要求】

n+1行：

第1行，n，表示学生个数；

第2至n+1行，每行输入一个学生信息，使用空格分隔输入；

第n+2行，需要删除学生的年龄

【输出要求】

若干行：

第1行：表头

第2至n+1行：学生信息结点。

第n+2：After Del:

第n+3行：表头

第n+4行以后：删除剩余的学生结点。

【输入样例】

4

101 Ma m 20

102 Li f 23

103 Zhang m 19

104 Wang m 19

19

【输出样例】

NO. name sex age

101 Ma m 20

102 Li f 23

103 Zhang m 19

104 Wang m 19

After Del::

NO. name sex age

101 Ma m 20

102 Li f 23

#### 5 链表合并

已有a,b两个链表，每个链表中的结点包括学号、成绩。要求把两个链表合并，并且按学号升序排列。

注：输入一个链表结束后，输入0表示结束；

注意输入输出格式。

【输入要求】

第1行至第n行，输入链表a的数据（学号 成绩）

第n+1行，输入0

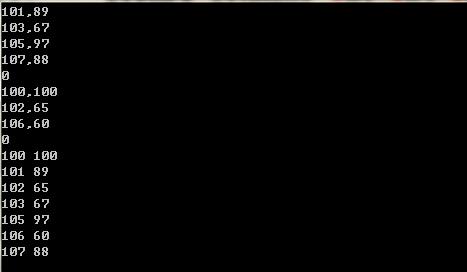
第n+2行至第m行，输入链表b的数据（学号 成绩）

第m+1行，输入0

【输出要求】

第1行至第k行，输出按照学号升序排好的数据（学号 成绩）

如图：



【输入样例】

101,89

103,67

105,97

107,88

0

100,100

102,65

106,60

0

【输出样例】

100 100

101 89

102 65

103 67

105 97

106 60

107 88