

太戈编程  
etiger.vip

# 信奥算法

# 自定义类型

**struct**  
结构体

# 超级英雄

输入n个超级英雄的名字和战斗力，储存后再输出。n<=10

输入样例：

3

Spiderman 80

Ironman 70

Antman 90

输出样例：

Spiderman: 80

Ironman: 70

Antman: 90

```
struct hero{  
    string name;  
    int power;  
};
```

创造新的数据类型hero  
记录英雄姓名和战斗力

```
hero h[10];
```

定义一个数组h  
包含10个hero类型的元素



# 超级英雄

```
1 #include<iostream>
2 #include<string>
3 using namespace std;
4 struct hero{
5     string name;
6     int power;
7 };
8 hero h[10];
9 int main(){
10     int n;
11     cin>>n;
12     for(int i=0;i<n;i++)
13         cin>>h[i].name>>h[i].power;
14     for(int i=0;i<n;i++)
15         cout<<h[i].name<<" : "<<h[i].power<<endl;
16     return 0;
17 }
```

自定义英雄类型  
hero

英雄的名字

英雄的战斗力

输入h数组里i号英雄的名字  
以及i号英雄的战斗力

# struct 结构体

C++的struct结构体  
能把不同类型的几个数据视为一个整体  
可以把各种类型的数据放在一起

结构体相当于自定义数据类型  
是一种复合数据类型。

# struct 结构体的赋值

```
struct hero{  
    string name;  
    int power;  
};  
hero a;
```

```
a={"spiderman",100};
```



```
a.name="spiderman";  
a.power=100;
```



# struct 结构体的赋值

```
struct hero{  
    string name="spiderman";  
    int power=100;  
};  
hero a;
```

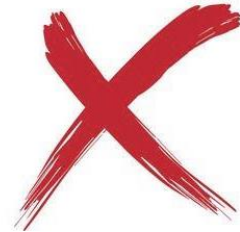


定义类型时不能赋值

# struct 结构体的输入

```
struct hero{  
    string name;  
    int power;  
};  
hero a;
```

```
cin>>a;
```



```
cin>>a.name>>a.power;
```





# struct 结构体的输出

```
struct hero{  
    string name;  
    int power;  
};  
hero a;
```

```
cout<<a;
```



```
cout<<a.name<<a.power;
```



# 自定义排序

# 英雄排序

输入n个超级英雄的名字和战斗力，按照战斗力从大到小排序，若战斗力相等按照名字字典序排序。n≤10

输入样例：

4

Superman 80

Spiderman 80

Ironman 70

Batman 90

输出样例：

Batman: 90

Spiderman: 80

Superman: 80

Ironman: 70



# 英雄排序

自定义英雄类型hero

```
5 struct hero{  
6     string name;  
7     int power;  
8 };  
9 hero h[10];
```

自定义比较规则cmp

compare缩写

判断英雄a是否  
排在英雄b前面

```
10 bool cmp(const hero&a, const hero&b){  
11     if(a.power > b.power) return true;  
12     if(a.power < b.power) return false;  
13     if(a.name < b.name) return true;  
14     return false;  
15 }
```

# 英雄排序

自定义英雄类型hero

```
5 struct hero{  
6     string name;  
7     int power;  
8 };  
9 hero h[10];
```

自定义比较规则cmp

compare缩写

判断英雄a是否  
排在英雄b前面

```
10 bool cmp(const hero&a, const hero&b){  
11     如果a的战斗力大于b的战斗力,那a是排在b前面  
12     如果a的战斗力小于b的战斗力,那a不排在b前面  
13     如果a的名字小于b的名字,那a是排在b前面  
14     返回: a不排在b前面  
15 }
```

先比战斗力

再比名字

# 英雄排序

```
1 #include<iostream>
2 #include<string>
3 #include<algorithm>
```

引入算法库

```
16 int main(){
17     int n;
18     cin>>n;
19     for(int i=0;i<n;i++)
20         cin>>h[i].name>>h[i].power;
21     sort(h,h+n,cmp);
22     for(int i=0;i<n;i++)
23         cout<<h[i].name<<": "<<h[i].power<<endl;
24     return 0;
25 }
```

请轮流翻译每一行

自定义排序

# 英雄排序

```
5 struct hero{  
6     string name;  
7     int power;  
8 };  
9 hero h[10];
```

为第10-14行  
翻译写注释

```
10 bool cmp(const hero&a, const hero&b){  
11     if(a.power > b.power) return true;  
12     if(a.power < b.power) return false;  
13     if(a.name < b.name) return true;  
14     return false;  
15 }
```

# 自定义排序步骤

```
#include<algorithm>
```

引入算法库

```
struct hero{string name; int power;};  
hero h[10];
```

自定义英雄类型hero

```
bool cmp(const hero&a,const hero&b){  
    if(a.power>b.power)return true;  
    if(a.power<b.power)return false;  
    if(a.name<b.name)return true;  
    return false;  
}
```

自定义排序规则cmp

```
sort(h,h+n,cmp);
```

执行自定义排序



# 易错点

```
struct hero(  
    string name;  
    int power;  
)  
hero a;
```

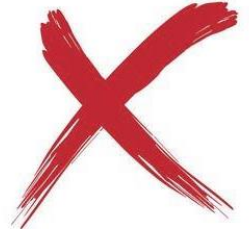


```
struct hero{  
    string name;  
    int power;  
};  
hero a;
```



# 易错点

```
bool cmp(hero a, hero b){  
    if(a.power>b.power)return true;  
    if(a.name<b.name)return true;  
}
```



```
bool cmp(const hero&a,const hero&b){  
    if(a.power>b.power)return true;  
    if(a.power<b.power)return false;  
    if(a.name<b.name)return true;  
    return false;  
}
```



# 成绩排序

输入n个学生学号,姓名和成绩,按照分数从高到低排序,成绩相等按照学号从小到大排序。 $n \leq 100$

输入样例:

4

3 Tom 80

2 Jerry 80

4 Mike 59

1 Peter 59

输出样例:

2 Jerry 80

3 Tom 80

1 Peter 59

4 Mike 59

自定义类型  
student

自定义比较规则  
cmp

先比成绩

再比学号

```
5 struct student{  
6     string name;  
7     int id,s;  
8 };  
9 student d[109];
```

自定义类型  
student

```
5 struct student{
6     string name;
7     int id,s;
8 };
9 student d[109];
10 bool cmp(const student&a,const student&b){
11     if(a.s>b.s) return true;
12     if(a.s<b.s) return false;
13     if(a.id<b.id) return true;
14     return false;
15 }
```

先比成绩  
再比学号

```
16 int main(){
17     int n;
18     cin>>n;
19     for(int i=0;i<n;i++)
20         cin>>d[i].id>>d[i].name>>d[i].s;
21     sort(d,d+n,cmp);
22     for(int i=0;i<n;i++){
23         cout<<d[i].id<<" ";
24         cout<<d[i].name<<" ";
25         cout<<d[i].s<<endl;
26     }
27     return 0;
28 }
```

# 排序步骤

算法可视化网址：  
**[visualgo.net/en/sorting](https://visualgo.net/en/sorting)**

Sorting is a very classic problem of reordering items of an array in a certain order.

There are many different sorting algorithms, each has its own advantages and limitations.

# 太戈编程**184**



## 自定义财宝类型

### `struct treasure`

请思考应该包含几个属性  
写出该类型的定义

```
4 struct treasure{  
5     int id,v,w;  
6 };
```

```
3  const int N=10009;
4  struct treasure{
5      int id,v,w;
6  };
7  treasure f[N];
8  bool cmp(const treasure &x, ) {
9      if(x.v>y.v) return 1;
10     if(x.v<y.v) return 0;
11     
12
13     if(x.id<y.id) return 1;
14     return 0;
15 }
```

自定义财宝类型  
struct treasure

```
19  int n;  
20  cin>>n;  
21  for(int i=0;i<n;i++)  
22      cin>>f[i].id>>f[i].v>>f[i].w;  
23    
24  for(int i=0;i<n;i++)  
25      cout<<<<endl;
```

太戈编程1497

## 自定义学生类型

### struct student

请思考应该包含几个属性  
写出该类型的定义

```
4 struct student{  
5     int id,score;  
6 };
```

```
17 int n;  
18 cin>>n;  
19 for(int i=0;i<n;i++){  
20     cin>>f[i].score;  
21     f[i].id=i+1;  
22 }  
23 sort(f,f+n,cmp);  
24 for(int i=0;i<n;i++){  
25     cout<<f[i].id<<" ";  
26     cout<<f[i].score<<endl;  
27 }
```

按照输入顺序分配学号

```
3  const int N=100009;  
4  struct student{  
5      int id,score;  
6  };  
7  student f[N];  
8  bool cmp(const student&a,const student&b){  
9  
10  
11  
12  
13 }
```

自定义学生类型  
struct student

# 太戈编程185



## 自定义英雄类型

### struct hero

请思考应该包含几个属性  
写出该类型的定义

```
4 struct hero{  
5     string name;  
6     int a,b,rp,sum;  
7 };
```

```
21 int main(){
22     int n;
23     cin>>n;
24     for(int i=0;i<n;i++) {
25         cin>>h[i].name>>h[i].a>>h[i].b>>h[i].rp;
26         h[i].sum=
27     }
28     sort(h,h+n,cmp);
29     for(int i=0;i<n;i++)
30         cout<<h[i].name<<endl;
31     return 0;
32 }
```

```
3  const int N=10009;
4  struct hero{
5      string name;
6      int a,b,rp,sum;
7  };
8  hero h[N];
9  bool cmp(const hero &x,const hero &y){
10
11
12      if(x rp>y rp) return 1;
13      if(x rp<y rp) return 0;
14      if(x a>y a) return 1;
15      if(x a<y a) return 0;
16      if(x.name<y.name) return 1;
17      return 0;
18  }
```

# 太戈编程

1497

184

185

拓展题

506,777,504,186