

单路和双路分支

夏秦

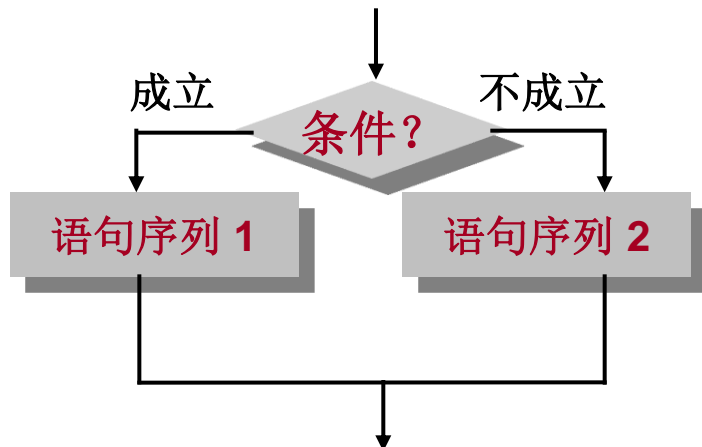
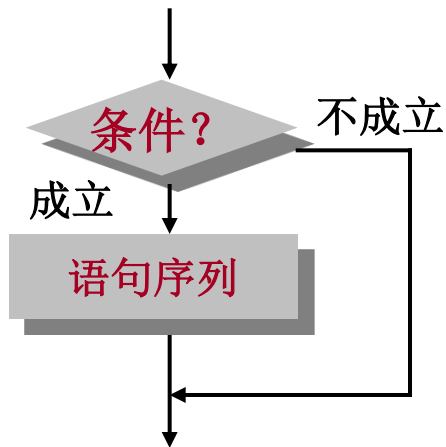
语句格式

▶ 单路分支

if (表达式)
语句序列

▶ 双路分支

if (表达式)
语句序列1
else
语句序列2



(1) 表示条件的表达式要能判断真假, 如 $a > 0$, $a \% 2 == 0$

(2) 当语句序列仅包含一条语句时, 可以省略花括号

例：比较两个数的大小

编写程序，找出并输出所输入两个实数中的较大数。

比较两个数大小的程序

```
#include<iostream>
using namespace std;
int main()
{
    double a,b,max;
    cin>>a>>b;
    max=a;
    if(a<b)
        max=b;
    cout<<max<<endl;
    return 0;
}
```

```
#include<iostream>
using namespace std;
int main()
{
    double a,b,max;
    cin>>a>>b;
    if(a<b)
        max=b;
    else
        max=a;
    cout<<max<<endl;
    return 0;
}
```

本例学到

- ▶ 1.单路分支语句

```
if(a<b)
```

```
    max=b;
```

- ▶ 2.双路分支语句

```
if(a<b)
```

```
    max=b;
```

```
else
```

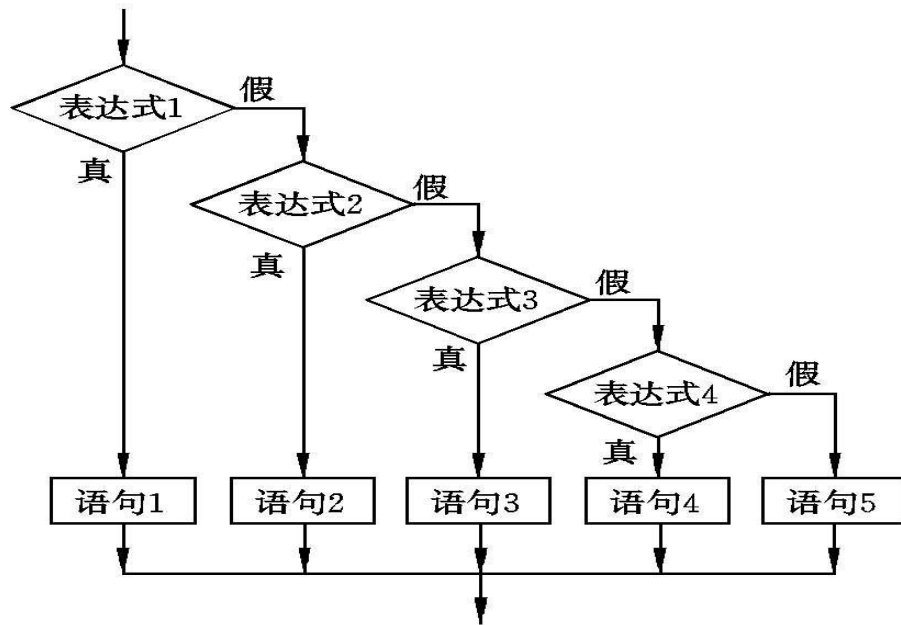
```
    max=a;
```

- ▶ 3.关系表达式作为条件

分支嵌套

if(表达式1) 语句1
else if(表达式2) 语句2
else if(表达式3) 语句3
else if(表达式4) 语句4
else 语句5

if(表达式1)
{ if (表达式2) 语句1 }
else 语句2 (内嵌if)



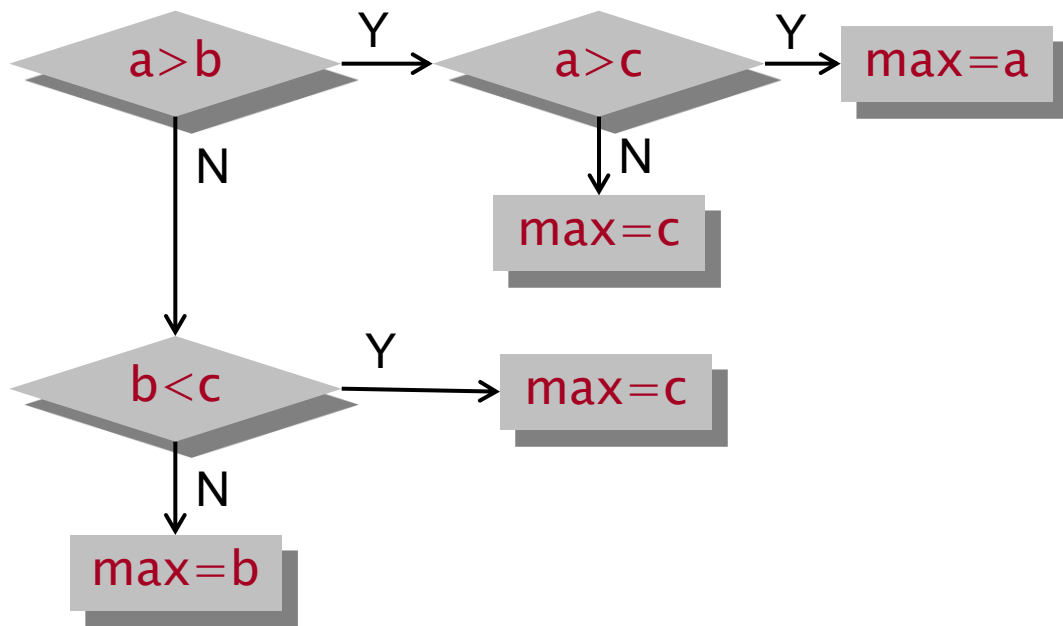
(1) else总是与它上面最近的if配对

(2) 如果if与else的数目不一样，可以加花括弧来确定配对关系

例：比较三个数的大小

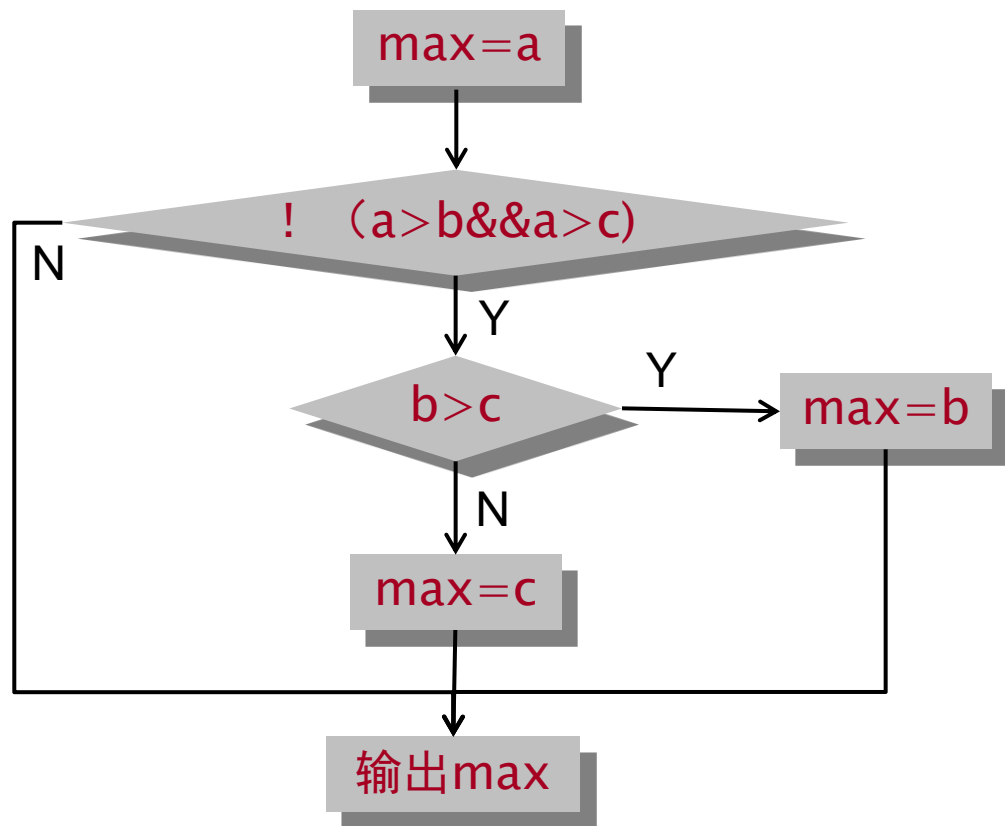
编写程序，找出并输出所输入三个实数中的最大数。

比较三个数大小的程序1



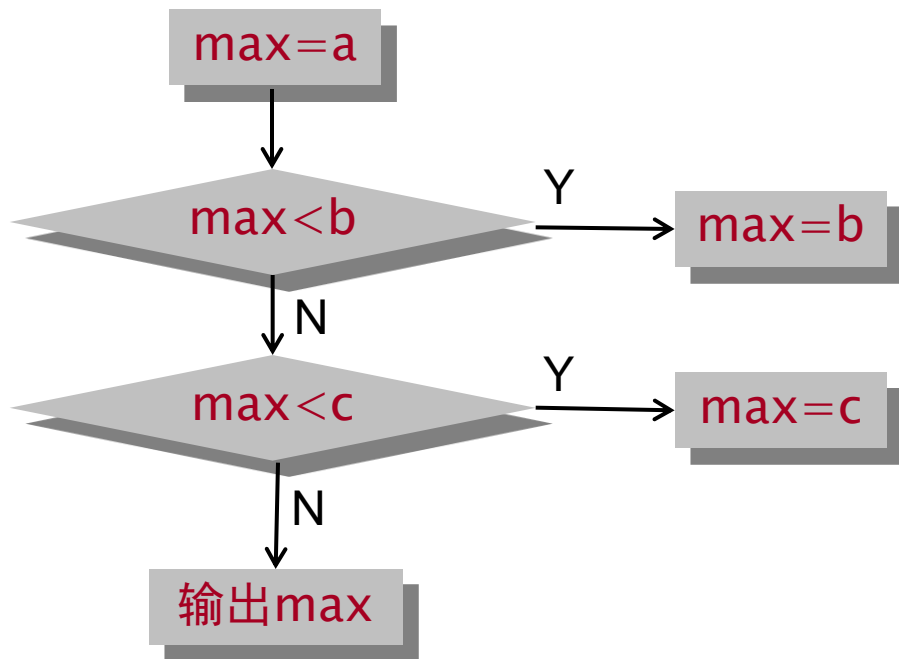
```
#include <iostream>
using namespace std;
int main()
{
    double a,b,c,max;
    cin >> a >> b >> c;
    if(a > b)
    {
        if(a > c)
            max = a;
        else
            max = c;
    }
    else
    {
        if(b < c)
            max = c;
        else
            max = b;
    }
    cout << max << endl;
    return 0;
}
```


比较三个数大小的程序2



```
#include<iostream>
using namespace std;
int main()
{ double a,b,c,max;
  cin>>a>>b>>c;
  max=a;
  if(!(a>b&& a>c))
  {   if(b>c)
      max=b;
      else
      max=c;}
  cout<<max<<endl;
  return 0;
}
```

比较三个数大小的程序3



```
#include<iostream>
using namespace std;
int main()
{ double a,b,c,max;
  cin>>a>>b>>c;
  max=a;
  if(max<b)
    max=b;
  if(max<c)
    max=c;
  cout<<max<<endl;
  return 0;
}
```

本例学到

▶ 1.分支嵌套

```
if!(a>b&&a>c)
{
    if(b>c)
        max=b;
    else
        max=c;
}
```

▶ 2.关系表达式和混合运算表达式作为条件

- (b>c)
- !(a>b&&a>c)

问题

- ▶ 在什么情况下，应该使用分支嵌套？
- ▶ 还有什么算法能够实现在三个数中，寻找最大数？