



[3] 分支结构程序设计

深入浅出程序设计竞赛
第 1 部分 – 语言入门
V 2021-02

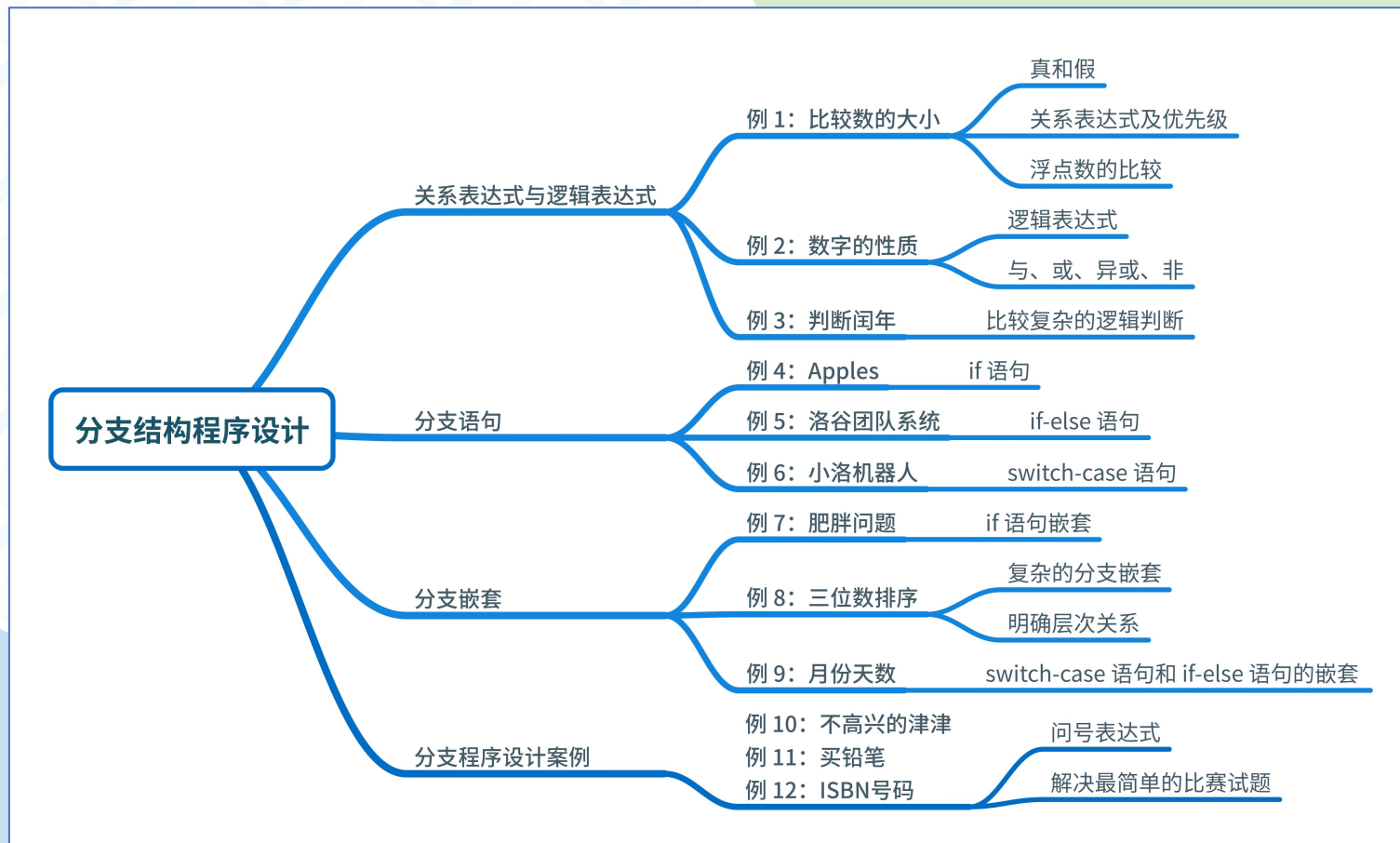
版权声明

本课件为《深入浅出程序设计竞赛 - 基础篇》的配套课件，版权归 洛谷 所有。所有个人或者机构均可免费使用本课件，亦可免费传播，但不可付费交易本系列课件。

若引用本课件的内容，或者进行二次创作，请标明本课件的出处。

- 其它《深基》配套资源、购买本书等请参阅：
<https://www.luogu.com.cn/blog/kkksc03/IPC-resources>
- 如果课件有任何错误，请在这里反馈
<https://www.luogu.com.cn/discuss/show/296741>

本章知识导图



第 3 章 分支结构程序设计

分支语句

请注意，为了更好的教学
这里安排的顺序和课本稍有差异

关系表达式与逻辑表达式

分支嵌套

分支程序设计案例

课后习题与实验

分支语句

程序的执行也不是一成不变的，往往会要求程序能够在不同的场合下有不同的动作。这时就需要在代码中使用条件语句来做出不同的选择。

请翻至课本 P36

if 语句

使用 if 语句 来控制程序在 指定条件 下需要做什么事情。

如果 条件成立（其值为真，或者 1）则 执行语句。

条件用 圆括号 包裹，执行语句用 大括号 包裹。

```
if (成立条件表达式) {  
    当条件成立时需要执行的语句;  
}
```

大括号里面那条语句相比于其他语句要往右移动了一点（前面加上空格或者Tab），这就是缩进。可以一目了然地知道那些语句是条件成立时才执行的，便于阅读。

Apples

例 3.4 (有所变动)

八尾勇吃掉了 $x(1 \leq x \leq 100)$ 个苹果。英语课上学到了 apple 这个词语，想用它来造句。

如果她吃了 1 个苹果，就输出 Today, I ate 1 apple. ;

如果她吃了不止一个苹果，别忘了 apple 这个单词后面要加上代表复数的 s。你能帮她完成这个句子吗？

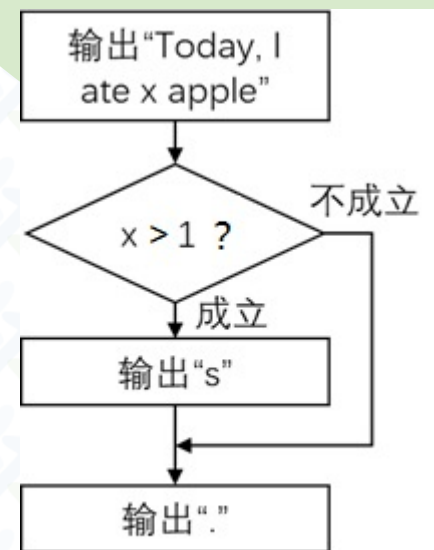
Apples

首先输入 x 数量，然后输出 Today, I ate x apple

使用 if 语句，如果条件成立（当 x 大于 1 时），输出复数的 s 。

条件执行语句加上缩进使代码更有层次感

```
#include <iostream>
using namespace std;
int main() {
    int x;
    cin >> x;
    cout << "Today, I ate " << x << " apple";
    if (x > 1) { // != 也可以, >=2 也可以
        cout << "s";
    }
    cout << "." << endl;
    return 0;
}
```



if-else 语句

配合上 **else**，就可以在条件成立的情况下做什么事情

否则，当条件**不成立**时做另外的一些事情

特别地，如果需要执行的语句只有一个**语句**，那么大括号可以不要

```
if (条件成立表达式) {  
    当条件成立时需要执行的语句;  
} else {  
    当条件不成立时需要执行的语句;  
}
```

```
if (条件成立表达式)  
    当条件成立时需要执行的【单条】语句;  
else  
    当条件不成立时需要执行的【单条】语句;
```

洛谷团队系统

例 3.5 (洛谷 P5713)

在洛谷上使用团队系统非常方便的添加自己的题目。

自己的电脑上配置：每题需要花费时间 5 分钟；

使用洛谷团队：每题只需 3 分钟，但是上传题目之前还需要一次性花费 11 分钟创建与配置团队。

现在要配置 $n(n \leq 100)$ 道题目，如果本地配置花费的总时间短，请输出 Local，否则输出 Luogu

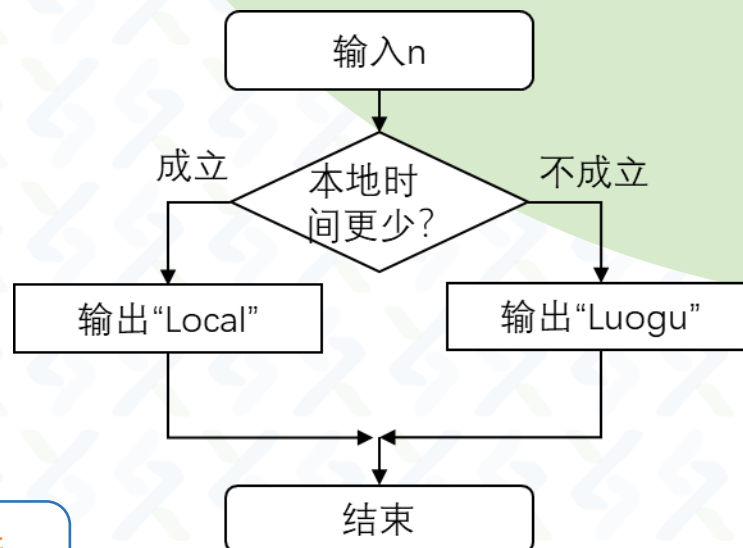
洛谷团队系统

本地上传： $5 \times n$ 上传洛谷： $11 + 3 \times n$

如果前者小于后者（if 条件成立），则选择在本地配置，否则选择洛谷上传（else 后面）

```
#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    if ((5 * n) < (11 + 3 * n)) {
        cout << "Local" << endl;
    } else {
        cout << "Luogu" << endl;
    }
    return 0;
}
```

注意合理使用换行和缩进
良好的代码规范更易理解！



switch-case

对于一些分叉很多（常数判断）的情况下，使用 if 会比较方便
使用了 `switch-case` 语句，判断一个变量是什么值，根据不同的值来进行操作

case 只能判定常量，执行完一个分支，在下一个 case 前加 `break`
如果 case 的所有情况都不符合，可以使用 `default`

```
switch (变量名) {  
    case 变量可能的情况1: 执行语句1; break;  
    case 变量可能的情况2: 执行语句2; break;  
    ...  
    default: 执行语句n;  
}
```

小洛机器人

例 3.6

提供以下指令（一个字符），机器人按照指令给出对应的回复：

- 输入G：打招呼，小洛会回复 Hello, my master!，还会在下一行加上一句 I'm Xiaoluo.
- 输入N：自我介绍，小洛只会回复 I'm Xiaoluo.
- 输入S：唱歌，小洛会哼唱 Teinei teinei teinei~
- 输入B或者Q：告别，小洛向你说 Bye bye!
- 其他任何字符：小洛无法理解，只能回复 Sorry...

小洛机器人

读入变量 opt, 然后使用 switch 语句判断各种情况

case 就是根据 opt 的值进入分支, 执行语句

使用 break 跳出分支, 否则, 继续执行下一个分支

如果 opt 的值不是 case 的情况, 就运行 default 后的语句

```
char opt; cin >> opt;
switch (opt) {
case 'G': cout << "Hello, my master!" << endl;
case 'N': cout << "I'm Xiaoluo." << endl; break;
case 'S': cout << "Teinei teinei teinei~" << endl; break;
case 'B': case 'Q':
    cout << "Bye bye!" << endl;
    break;
default: cout << "Sorry..." << endl;
}
```

关系表达式与逻辑表达式

虽然现实生活中很多事情真假难辨，但是在计算机中真假是黑白分明的。计算机的世界由 0 和 1 组成，0 代表假，1 代表真。

请翻至课本 P33

关系表达式、优先级

判断两个元素的关系：

- 大于 $>$
- 小于 $<$
- 等于 $==$ (不是 $=$)
- 大于等于 $>=$
- 小于等于 $<=$
- 不等于 $!=$

例如表达式 $a > b$ ，若 a 的确大于 b ，该表达式为 1 (真) 否则为 0 (假)

优先级高的部分先进行计算
优先级 (从高到低)：

()	括号
* / %	乘除表达式
+ -	加减表达式
< > >= <=	大于小于
== !=	等于不等于

判断数字

例 3.1

输入两个整数 a 和 b ，我们想知道：

- a 是否大于 b ？
- a 是否小于等于 b ？
- a 是否不等于 b ？

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

假设 a 为 2， b 为 3，那么很显然 $a > b$ 不成立， $a \leq b$ 成立， $a \neq b$ 成立。程序的输出是 0 1 1。

注意如果没有括号的话会编译错误。

浮点数精度误差

浮点数可能会产生精度误差

所以一般不用 `==` 来判断两个浮点数是否相等

正确的方式：比较这两个数的差值是否小于一定程度

假设 `fabs(a-b)<1e-6` 成立，就可以认为浮点数 `a` 和 `b` 相等。

注意要用 `<cmath>` 头文件

```
double a, b;
```

```
.....
```

```
if (a == b) // 这么写是错误的！会造成浮点数误差！
```

```
if (fabs(a - b) < 1e-6) // 正确的写法，注意 <cmath>
```

逻辑表达式

多个条件复合成一个条件进行判断。

- 与： $a \& \& b$ ，当这两个条件都为真，其为 1
- 或： $a \parallel b$ ，当两个条件至少有一个为真，其为 1，否则 0
- 异或： $a \wedge b$ ，当两个条件刚好一个为真，其为 1，否则 0
- 非： $!a$ ，当 a 为假时，其为 1，否则 0

具体的 0-1 真值表见课本 P35

数的性质

例 3.2 (洛谷 P5710)

一些数字可能拥有以下的性质：

- 性质 1：是偶数；
- 性质 2：大于 4 且不大于 12。

小A 喜欢这两个性质同时成立的数字；Uim 喜欢这至少符合其中一种性质的数字；八尾勇喜欢刚好有符合其中一个性质的数字；正妹喜欢不符合这两个性质的数字。

现在输入一个数字 $x(0 \leq x \leq 100)$ ，要求输出这 4 个人是否喜欢这个数字，如果喜欢则输出 1，否则输出 0，用空格分隔。

数的性质

p1 和 p2 代表是否满足这两个性质：

- p1：是偶数；（ $x \% 2 == 0$ ，或者 $!x \% 2$ ）注意：非零为真
- p2：大于 4 且不大于 12。（ $4 < x \ \&\& \ x \leq 12$ ）

两个性质同时成立？ $p1 \ \&\& \ p2$

两个性质至少一个成立？ $p1 \ || \ p2$

两个性质刚好一个成立？ $p1 \ ^ \ p2$

两个性质同时不成立？

$!p1 \ \&\& \ !p2$ 或 $!(p1 \ || \ p2)$

```
int x; bool p1, p2;
cin >> x;
p1 = x % 2 == 0;
p2 = 4 < x && x <= 12;
cout << (p1 && p2) << ' ';
cout << (p1 || p2) << ' ';
cout << (p1 ^ p2) << ' ';
cout << (!p1 && !p2);
// cout << !(p1 || p2);
```

逻辑运算注意点

使用逻辑运算符构造出更加复杂的逻辑表达式

可以加上括号使条件变得清晰

不要把 `==` 错打成 `=` ← 初学者超级常犯的错误

注意优先级，逻辑/比较运算优先级较低

(、)	!、-(负号)、 ++、--	*、/、 %	+、-(加 减运算)	<<、>>(左 右位移)	<、>、 >=、<=	==、!=	&&	
← 优先度高				优先度低 →				

闰年判断

例 3.3 (洛谷 P5711)

输入一个年份（大于 1582 的整数），判断这一年是否为闰年，如果是输出 1，否则输出 0。

- 被 4 整除 (p1) 是闰年，除非
- 被 100 整除 (p2) 不是闰年，除非
- 被 400 整除 (p3) 又是闰年

稍微有点难？

闰年判断

输入一个年份（大于 1582 的整数），判断这一年是否为闰年，如果是输出 1，否则输出 0。

- 被 4 整除（ $p1 : x \% 4 == 0$ ）是闰年，除非
- 被 100 整除（ $p2 : x \% 100 == 0$ ）不是闰年，除非
- 被 400 整除（ $p3 : x \% 400 == 0$ ）又是闰年

$p3$ 成立必闰年， $p3$ 不成立时， $p1$ 成立且 $p2$ 不成立为闰年

$$p3 \quad || (\quad p1 \quad \&\& \quad !p2 \quad)$$
$$(x \% 400 == 0) || (x \% 4 == 0) \&\& (x \% 100 != 0)$$

```
int x; bool p;  
cin >> x;  
p = (x % 400 == 0) || (x % 4 == 0) && (x % 100 != 0);  
//p = !(x % 400) || !(x % 4) && x % 100;  
cout << p << endl;
```


短路判断

课本里没提到的补充内容

计算表达式时，如果只计算部分内容就可以确保一个表达式的结果，那么剩余部分计算机就不会继续运算下去了。

例如：

- $a || (b \& \& !c)$ ，当 a 为真，表达式肯定也是真，不用继续算
- $a \& \& (b || c || d || e || f)$ ，当 a 为假，表达式肯定为假
- $x > 0 \& \& \text{sqrt}(x) - \text{round}(\text{sqrt}(x)) < 1e-6$ ，当 x 是负数时，后面的就不会继续运算，也不会造成 RE。

分支嵌套

有时候条件并不是只有两个分支，我们经常会遇到比较复杂的分支

请翻至课本 P40

if 嵌套

如果需要分成三个以上分支、或者一个分支中还需要继续分类，使用**条件嵌套**。

if 语句可以嵌套，组成更复杂的条件分支。例如下面两种：

```
if (条件 1) {  
    条件 1 成立执行语句;  
} else if(条件 2) {  
    条件 1 不成立、条件 2 成立执行语句;  
} else {  
    条件 1 和 2 均不成立执行语句;  
}
```

```
if (条件 1) {  
    条件 1 成立执行语句;  
} else {  
    if(条件 2) {  
        条件1不成立、条件2成立执行语句;  
    } else {  
        条件 1 和 2 均不成立执行语句;  
    }  
}
```

肥胖问题

例 3.7 (洛谷P5714)

BMI 指数算法是 $\frac{m}{h^2}$ ($40 \leq m \leq 120, 1.4 \leq h \leq 2.0$), 其中 m 是指体重 (千克), h 是指身高 (米)。不同体型范围与判定结果如下:

- 小于 18.5 : 体重过轻, 输出 Underweight ;
- 大于等于 18.5 且小于 24 : 正常体重, 输出 Normal ;
- 大于等于 24 : 肥胖, 不仅要输出 BMI 值 (使用 cout 的默认精度), 还要输出 Overweight ;

现在给出体重和身高数据, 需要根据 BMI 指数判断体型状态并输出对应的判断

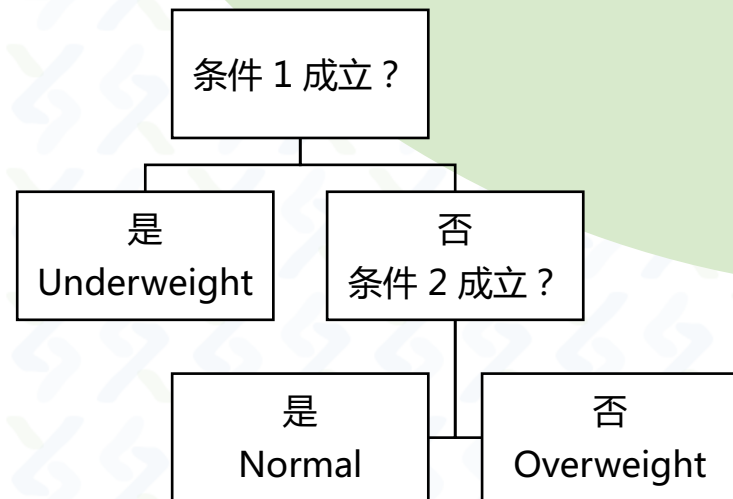
肥胖问题

条件 1：小于 18.5，Underweight

如果不满足条件 1，则条件 2：小于 24，Normal

如果条件 1 和 2 都不满足：Overweight

```
double m, h, BMI;  
cin >> m >> h;  
BMI = m / h / h;  
if (BMI < 18.5)  
    cout << "Underweight";  
else if (BMI < 24)  
    cout << "Normal";  
else {  
    cout << BMI << endl;  
    cout << "Overweight" << endl;  
}
```



逗号表达式

将两个不同的表达式写在了一起，变成了一条语句，变成了逗号表达式。有时可以使语句变得精炼

```
//正确写法 1 使用大括号包括多个语句
if (t1 + t2 > maxtime) {
    maxtime = t1 + t2; maxday = 1;
}
//正确写法2使用逗号表达式链接多个语句成为一个语句
if (t1 + t2 > maxtime)
    maxtime = t1 + t2, maxday = 1;
//错误写法，第二个语句不在if管辖范围内
if (t1 + t2 > maxtime)
    maxtime = t1 + t2; maxday = 1;
```

三位数排序

例 3.8 填空练习 (洛谷 P5715)

给出三个整数 a, b, c ($0 \leq a, b, c \leq 100$), 把这三个整数从小到大排序。

解法 1: 将 3 个整数从小到大排列, 可能有 $[a \ b \ c] [a \ c \ b] [b \ a \ c] [b \ c \ a] [c \ a \ b] [c \ b \ a]$ 这六种排列。

枚举这 6 种排列关系 (小中大), 就可依次判断属于哪种情况了。

```
int a, b, c;
scanf("%d%d%d", &a, &b, &c);
if (a <= b && b <= c) printf("%d %d %d\n", a, b, c);
else if (a <= c && c <= b) printf("%d %d %d\n", a, c, b);
else if (b <= a && a <= c) printf("%d %d %d\n", b, a, c);
else if (b <= c && c <= a) printf("%d %d %d\n", b, c, a);
else if (c <= a && a <= b) printf("%d %d %d\n", c, a, b);
else /*if (c <= b && b <= a)*/ printf("%d %d %d\n", c, b, a);
```

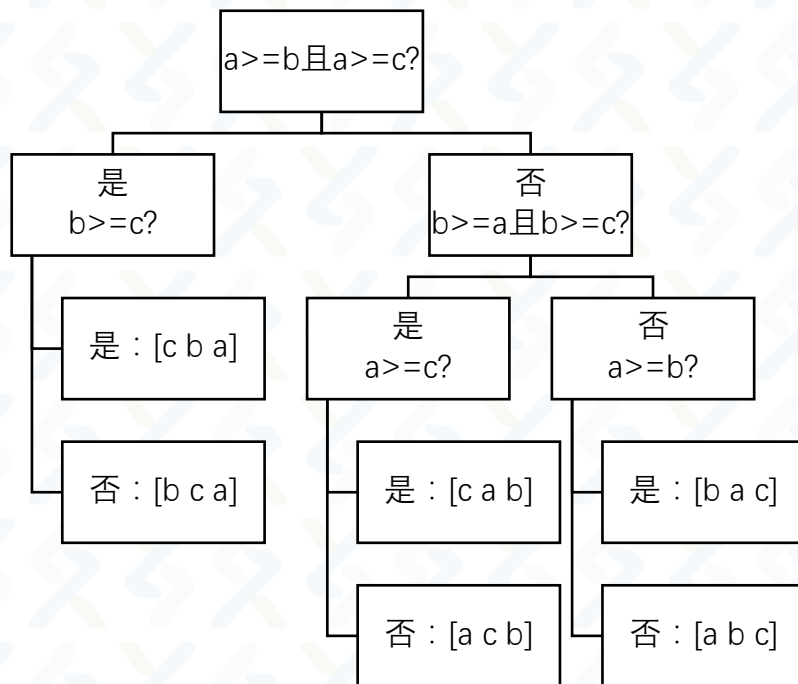
三位数排序

解法 2：首先拎出最大的一个数字，分为 3 种情况：a 最大、b 最大和 c 最大。对于每一种情况，比较剩下两个变量的大小关系。

这样还是有 6 种情况。注意层级问题（省了大括号）

```
int a, b, c;
scanf("%d%d%d", &a, &b, &c);
if (a >= b && a >= c)
    if (b >= c) printf("%d %d %d\n", c, b, a);
    else printf("%d %d %d\n", b, c, a); // 这个else搭配哪个if呢？
else if (b >= a && b >= c)
    if (a >= c) printf("%d %d %d\n", c, a, b);
    else printf("%d %d %d\n", a, c, b);
else // if (c >= a && c >= b) 本句可加可不加
    if (a >= b) printf("%d %d %d\n", b, a, c);
    else printf("%d %d %d\n", a, b, c);
return 0;
```


else 的配对方式和括号



```
#include <stdio>
using namespace std;
int main() {
    int a, b, c;
    scanf("%d%d%d", &a, &b, &c);
    if (a >= b && a >= c)
        if (b >= c) printf("%d %d %d\n", c, b, a);
        else printf("%d %d %d\n", b, c, a);
    else if (b >= a && b >= c)
        if (a >= c) printf("%d %d %d\n", c, a, b);
        else printf("%d %d %d\n", a, c, b);
    else // if (c >= a && c >= b) 本句可加可不加
        if (a >= b) printf("%d %d %d\n", b, a, c);
        else printf("%d %d %d\n", a, b, c);
    return 0;
}
```

else跟哪个if配对呢？

无括号的话，按照层次进行else-if配对

三位数排序

解法 3：三次交换（书中没讲）

确保第一个数是最小的：如果第一个数大于第二个数则他们交换；
如果第二个数小于第三个数则他们交换

然后确认第二个数是第二小的：如果第二个数大于第三个数则他们交换；这里使用了逗号表达式，也可以用分号加大括号

```
int a, b, c, p;  
scanf("%d%d%d", &a, &b, &c);  
if (a > b)  
    p = a, a = b, b = p;  
if (a > c)  
    p = a, a = c, c = p;  
if (b > c)  
    p = b, b = c, c = p;  
printf("%d %d %d", a, b, c);
```

分支程序设计案例

其中的一些题目是各类比赛的真题（但是现在遇到的都很简单，不用担心！）

请翻至课本 P43

不高兴的津津

例 3.10 （洛谷 P1085）

津津除了上学之外，还要参加各科课外补习班。

津津一天学习超过8小时就会不高兴，上得越久越不高兴。

已知津津下周每天的上学和补习班学习（输入 7 行，每行 2 个数字），看看下周她会不会不高兴；如果会的话，哪天最不高兴。

提示：定义变量 `maxtime` 记录最长的学习时间

`maxday` 变量用于记录星期几的时候打破纪录的

每一天都要判断一下是不是超了

不高兴的津津

提示：定义变量 `maxtime` 记录最长的学习时间，`maxday` 变量用于记录星期几的时候打破纪录。每一天都要判断一下是不是超了。

之后学习循环语句，就不需要复制这么多遍了

```
int t1, t2, maxtime = 8, maxday = 0;
cin >> t1 >> t2;
if (t1 + t2 > maxtime) maxtime = t1 + t2, maxday = 1;
cin >> t1 >> t2;
if (t1 + t2 > maxtime) maxtime = t1 + t2, maxday = 2;
cin >> t1 >> t2;
if (t1 + t2 > maxtime) maxtime = t1 + t2, maxday = 3;
cin >> t1 >> t2;
if (t1 + t2 > maxtime) maxtime = t1 + t2, maxday = 4;
cin >> t1 >> t2;
if (t1 + t2 > maxtime) maxtime = t1 + t2, maxday = 5;
cin >> t1 >> t2;
if (t1 + t2 > maxtime) maxtime = t1 + t2, maxday = 6;
cin >> t1 >> t2;
if (t1 + t2 > maxtime) maxtime = t1 + t2, maxday = 7;
cout << maxday;
```

注释

单行注释：`// 斜杠后面的一行都会被忽略`

多行注释：`/* 从这里
开始的很多行
都是注释，内容会被忽略
直到结尾 */`

程序会忽略掉注释的内容，但是写注释可以帮助自己记录思路或者，将一部分程序给“屏蔽”而不执行

问号表达式

问号表达式的形式是 $S1?S2:S3$

如果 $S1$ 条件成立，那么这个表达式的值是 $S2$ ，否则是 $S3$ 。

例如：`b=a==1?2:3`

意思是 $b = (a==1) ? 2 : 3$

也就是当 $a==1$ 成立时， $(a==1) ? 2 : 3$ 的值是 2，即 b 赋 2；否则 $(a==1) ? 2 : 3$ 的值是 3， b 赋值 3。

买铅笔

例 3.11 (洛谷 P1909)

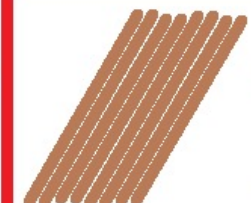
要买 n 支铅笔，商店一共有 3 种包装的铅笔，不同包装内的铅笔数量和总价可能不同且已知。只买同一种包装的铅笔。

由于铅笔的包装不能拆开，可能需要购买超过 n 支铅笔才够。

请问在商店每种包装的数量都足够的情况下，要买够至少 n 支铅笔最少需要花费多少钱。所有输入数据不超过 10000。

 2支装
2元
29包 58元

 50支装
30元
2包 60元

 30支装
27元
2包 54元

买铅笔

计算出每种类型的笔需要买几包

假设需要 n 支笔，每包有 $n1$ 支，那么如果 n 可以整除 $n1$ （也就是 $n \% n1 == 0$ ），那么就需要 $n / n1$ 包；如果不能整除，那么还需要多买一包，也就是 $n / n1 + 1$ 包。

乘上单价就是总价格，通过打擂台的方式比较哪种包装笔最便宜。

```
int n, n1, n2, n3, p1, p2, p3, t1, t2, t3, total;
cin >> n >> n1 >> p1 >> n2 >> p2 >> n3 >> p3;
t1 = !(n % n1) ? n / n1 * p1 : (n / n1 + 1) * p1;
t2 = !(n % n2) ? n / n2 * p2 : (n / n2 + 1) * p2;
t3 = !(n % n3) ? n / n3 * p3 : (n / n3 + 1) * p3;
total = t1; // 假设第一种是最省钱的方案
if (t2 < total) total = t2;
if (t3 < total) total = t3;
cout << total << endl;
```

课后习题与实验

学而时习之，不亦说乎。学而不思则罔，思而不学则殆。——孔子

请翻至课本 P45

判断练习

习题 3.1

当 $a = 3, b = 4, c = 5$, 判断以下表达式是否成立？

- $a < b \parallel b > c \parallel a > b$
- $a > c \parallel b > a \&\& c > b$
- $b - a == c - b$
- $a * b - c > a * c - b \parallel a * b + b * c == b * b * (c - a)$

判断练习

习题 3.1 – 答案

当 $a = 3, b = 4, c = 5$ ，判断以下表达式是否成立？

- $a < b || b > c || a > b$
成立， $3 < 4 || 4 > 5 || 3 > 4$ ，真||假||假
- $a > c || b > a \&\& c > b$
成立， $3 > 5 || (4 > 3 \&\& 5 > 4)$ ，假||(真&&真)
- $b - a == c - b$
成立， $1 == 1$
- $a * b - c > a * c - b || a * b + b * c == b * b * (c - a)$
成立， $3 * 4 - 5 > 3 * 5 - 4 || 3 * 4 + 4 * 5 == 4 * 4 * (5 - 3)$
 $7 > 11 || 32 == 32$

判断练习

习题 3.2

当 $a=1, b=0, c=1$ ，判断以下表达式是否成立？

- $!a||!b$
- $(a\&\&!a)||(b||!b)$
- $a\&\&b\&\&c||!a||!c$
- $a\&\&(b\&\&c||a\&\&c)$
- $!b\&\&(c\&\&(a\&\&(!c||(!b||(!a))))$

判断练习

习题 3.2 – 答案

当 $a=1, b=0, c=1$ ，判断以下表达式是否成立？

- $!a || !b$ 成立, $0 || 1$
- $(a \& \& !a) || (b || !b)$ 成立, $(b || !b)$ 永真
- $a \& \& b \& \& c || !a || !c$ 不成立
- $a \& \& (b \& \& c || a \& \& c)$ 成立, $1 \& \& (0 \& \& 1 || 1 \& \& 1)$
- $!b \& \& (c \& \& (a \& \& (!c || (!b || (!a))))))$ 成立

总结

if 语句、if-else 语句

如果给定的表达式为真，那么执行指定语句
否则（else）执行另外的语句

switch-case 分支语句

判断某个表达式的值在几个常量的值
每个分支要加 break 退出，default 其余情况

条件语句嵌套

可以多个 else 多个分支，或者分支中还有分支
注意复杂的分支语句，以及大括号的用法

缩进、逗号表达式、注释、问号表达式

让编程更容易的一些窍门

总结

关系表达式

真 (1) 假 (0)

大于小于等于不等于，注意优先级

逻辑表达式

与 (&&) : 同时为真则为真

或 (||) : 至少有一个为真则为真

异或 (^) : 刚好有一个为真则为真

非 (!) : 颠倒黑白

浮点数精度误差、短路判断

作业

习题 3.3 (见课本 P46)

对于整形变量 x ，写出判断这些性质对应的表达式

可以编写完整的程序，然后自己代入几个值检查是否正确

习题 3.4 小玉家的电费 (洛谷 P1422)

某地用电标准如下：

- 月用电量在 150 千瓦时及以下部分每千瓦时 0.4463 元
- 月用电量在 151~400 千瓦时的部分每千瓦时 0.4663 元
- 月用电量在 401 千瓦时及以上部分每千瓦时 0.5663 元

已知当月用电量（整数且不大于 10000），根据电价规定，计算出应交的电费应该是多少。保留一位小数

课后扩展

习题 3.7 陶陶摘苹果（洛谷 P1046，NOIP2005 普及组）

一棵苹果树结出 10 个苹果。陶陶要摘，她有个30厘米高的板凳。
当她不能直接用手摘到苹果的时候，就会踩到板凳上再试试。
假设她碰到苹果，苹果就会掉下来。

现在已知 10 个苹果到地面的高度，以及陶陶把手伸直的时候能够达到的最大高度。请帮陶陶算一下她能够摘到的苹果的数量。

100 200 150 140 129 134 167 198 200 111
110

5

提示：定义 10 个苹果的高度，然后分别读入，分别判断

课后扩展

习题 3.8 三角形分类 (洛谷 P5717)

给出三条线段 a, b, c 的长度，拼成三角形，它能是什么三角形呢？

- 如果三条线段不能组成一个三角形，输出 Not triangle ；
- 如果是直角三角形，输出 Right triangle ；
- 如果是锐角三角形，输出 Acute triangle ；
- 如果是钝角三角形，输出 Obtuse triangle ；
- 如果是等腰三角形，输出 Isosceles triangle ；
- 如果是等边三角形，输出 Equilateral triangle。

如果三角形符合以上多个条件，请分别输出，并用换行符隔开。

参考阅读材料

以下内容限于课件篇幅未能详细阐述。如果学有余力，可自行翻阅课本作为扩展学习。

- P42 例 3.9：switch-case 和 if-else 语句的嵌套
- P45 例 3.12：解决简单的算法竞赛真题方法
- 习题 3.9。