

嵌套的if

找三个数中的最大

```
int a,b;

printf("请输入两个整数: ");
scanf("%d %d", &a, &b);

int max = 0;
if ( a > b ) {
    max = a;
} else {
    max = b;
}

printf("大的那个是%d\n", max);
```

- 如果要求输入三个整数，输出其中最大的，应该怎么做？

```

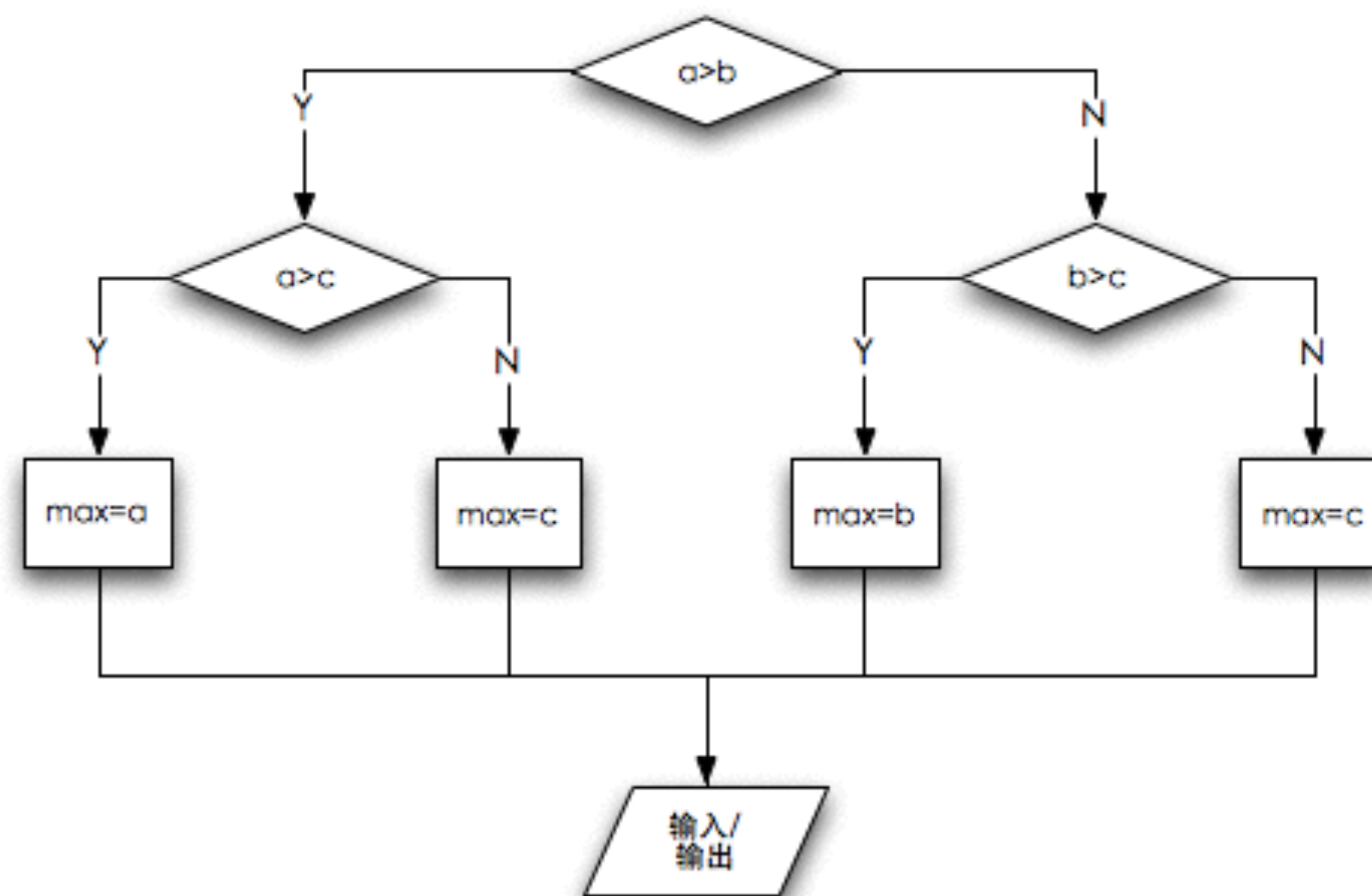
int a,b,c;
scanf("%d %d %d", &a, &b, &c);

int max = 0;

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

printf("The max is %d\n", max);

```




```
int a,b,c;
scanf("%d %d %d", &a, &b, &c);

int max = 0;

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

printf("The max is %d\n", max);
```

```
int a,b,c;
scanf("%d %d %d", &a, &b, &c);

int max = 0;

if ( a>b )
    if ( a>c )
        max = a;
    else
        max = c;
else
    if ( b>c )
        max = b;
    else
        max = c;

printf("The max is %d\n", max);
```

嵌套的判断

- 当if的条件满足或者不满足的时候要执行的语句也可以是一条if或if-else语句，这就是嵌套的if语句

```
if ( code == READY )  
    if ( count <20 )  
        printf("一切正常\n");  
    else  
        printf("继续等待\n");
```

else的匹配

- else总是和最近的那个if匹配

```
if ( code == READY ) {  
    if ( count < 20 )  
        printf("一切正常\n");  
} else  
    printf("继续等待\n");
```

缩进

- 缩进格式不能暗示else的匹配

```
if ( code == READY )  
    if ( count < 20 )  
        printf("一切正常\n");  
else  
    printf("继续等待\n");
```

嵌套的if

```
if ( gameover == 0 )  
    if ( player2move == 2 )  
printf("Your turn\n");
```



```
if ( gameover == 0 )  
    if ( player2move == 2 )  
        printf("Your turn\n");  
    else  
        printf("My turn\n");
```

```
if ( gameover == 0 )  
    if ( player2move == 2 )  
        printf("Your turn\n");  
    else  
        printf("My turn\n");  
else  
    printf("GAME OVER\n");
```

```
if ( gameover == 0 )  
    if ( player2move == 2 )  
        printf("Your turn\n");  
else  
    printf("GAME OVER\n");
```

```
if ( gameover == 0 ) {  
    if ( player2move == 2 )  
        printf("Your turn\n");  
} else  
    printf("GAME OVER\n");
```

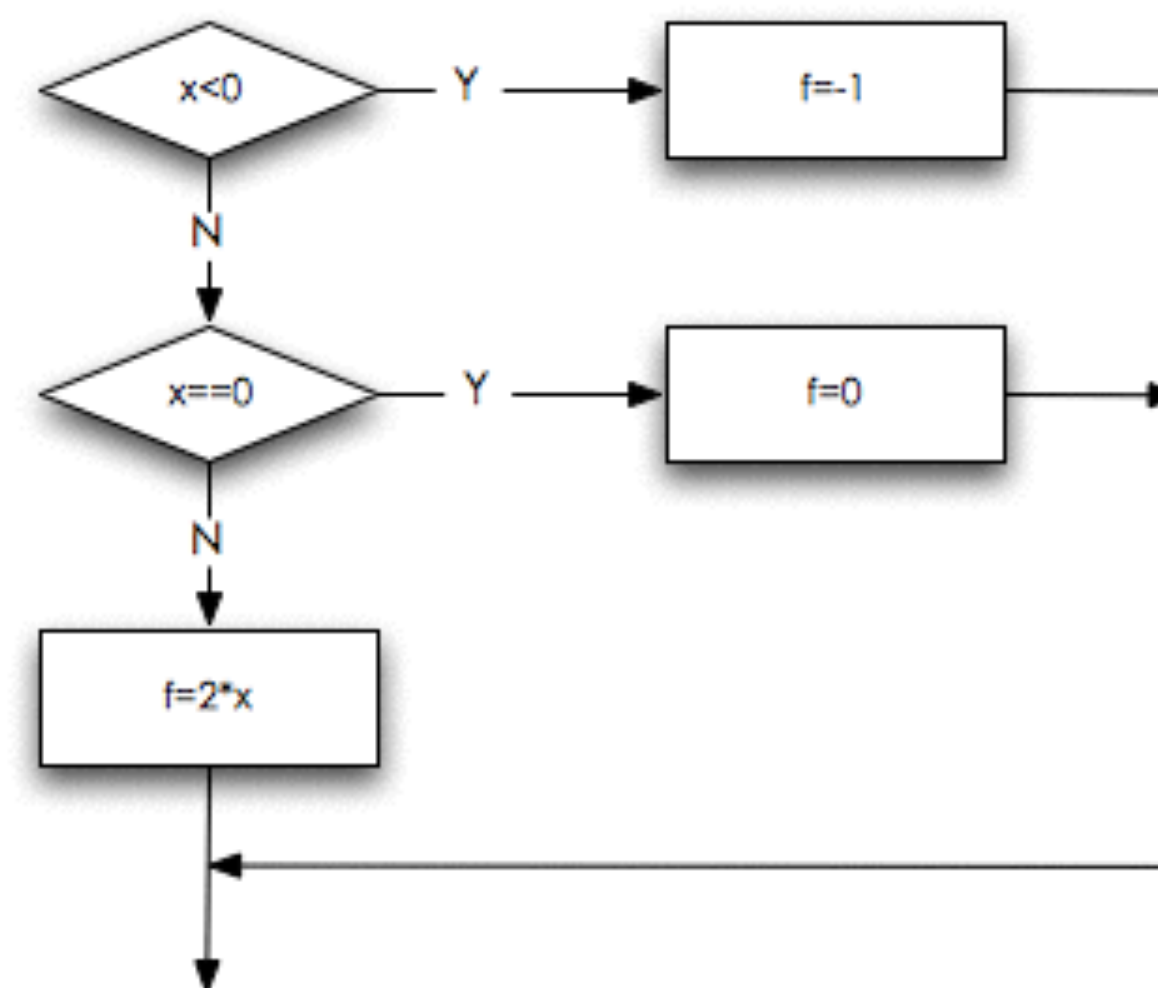
tips

- 在if或else后面总是用{}
- 即使只有一条语句的时候

级联的if-else if

分段函数

```
if ( x < 0 ) {  
    f = -1;  
} else if ( x == 0 ) {  
    f = 0;  
} else {  
    f = 2 * x;  
}
```



级联的if-else if

```
if ( exp1 )
```

```
    st1;
```

```
else if ( exp2 )
```

```
    st2;
```

```
else
```

```
    st3;
```

```
int f;  
if ( x < 0 ) {  
    f = -1;  
} else if ( x == 0 )  
    f = 0;  
} else {  
    f = 2 * x;  
}  
printf("%d", f);
```

```
if ( x < 0 ) {  
    printf("%d", -1);  
} else if ( x == 0 ) {  
    printf("%d", 0);  
} else {  
    printf("%d", 2 * x);  
}
```

单一出口

if语句常见错误

if语句常见的错误

- 忘了大括号
- if后面的分号
- 错误使用==和=
- 使人困惑的else

忘了大括号

```
if ( age > 60 )  
    salary = salary * 1.2;  
    printf("%f",sal
```

```
if ( age > 60 )  
{  
    salary = salary * 1.2;  
    printf("%f",salay);  
}
```

- 永远在if和else后面加上大括号，即使当时后面只有一条语句

if后面的分号

```
if ( age > 60 );  
{  
    salary = salary * 1.2;  
    printf("%f",salay);  
}
```

错误使用==和=

- if只要求()里的值是零或非零

```
if ( a = b )  
{  
    printf("A=B");  
}
```

代码风格

- 在if和else之后必须加上大括号形成语句块；
- 大括号内的语句缩进一个tab的位置；

风格是三观...

```
if ( x < 0 ) {  
    f = -1;  
} else if ( x == 0 ) {  
    f = 0;  
} else {  
    f = 2 * x;  
}
```

```
if ( x < 0 )  
{  
    f = -1;  
} else if ( x == 0 )  
{  
    f = 0;  
} else  
{  
    f = 2 * x;  
}
```

```
if ( x < 0 )  
{  
    f = -1;  
}  
else if ( x == 0 )  
{  
    f = 0;  
}  
else  
{  
    f = 2 * x;  
}
```

- 在if和else之后必须加上大括号形成语句块;
- 大括号内的语句缩进一个tab的位置;

多路分支

switch-case

```
if ( type==1 )  
    printf("你好");  
else if ( type==2 )  
    printf("早上好");  
else if ( type==3 )  
    printf("晚上好");  
else if ( type==4 )  
    printf("再见");  
else  
    printf("啊，什么啊? ");
```

```
switch ( type ) {  
    case 1:  
        printf("你好");  
        break;  
    case 2:  
        printf("早上好");  
        break;  
    case 3:  
        printf("晚上好");  
        break;  
    case 4:  
        printf("再见");  
        break;  
    default:  
        printf("啊，什么啊? ");  
}
```

switch-case

```
switch ( 控制表达式 ) {  
    case 常量:  
        语句  
        .....  
    case 常量:  
        语句  
        .....  
    default:  
        语句  
        .....  
}
```

- 控制表达式只能是整数型的结果
- 常量可以是常数，也可以是常数计算的表达式
- 根据表达式的结果，寻找匹配的case，并执行case后面的语句，一直到break为止
- 如果所有的case都不匹配，那么就执行default后面的语句；如果没有default，那么就什么都不做

break

```
switch ( type )
{
case 1:
case 2:
    printf("你好\n");
    break;
case 3:
    printf("晚上好\n");
case 4:
    printf("再见\n");
    break;
default:
    printf("啊，什么啊? \n");
    break;
}
```

- **switch**语句可以看作是一种基于计算的跳转，计算控制表达式的值后，程序会跳转到相匹配的**case**（分支标号）处。分支标号只是说明**switch**内部位置的路标，在执行完分支中的最后一条语句后，如果后面没有**break**，就会顺序执行到下面的**case**里去，直到遇到一个**break**，或者**switch**结束为止。

{ Programming Ability Test

主页

题目集

基本信息

题目列表

提交列表

排名

帮助

分支-06. 成绩转换(15)

本题要求编写程序将一个百分制成绩转换为五分制成绩。转换规则：

- 大于等于90分为A；
- 小于90且大于等于80为B；
- 小于80且大于等于70为C；
- 小于70且大于等于60为D；
- 小于60为E。

输入格式：

输入在一行中给出1个整数的百分制成绩。

输出格式：

输入样例：

90

输出样例：

A

```
//printf("输入成绩 (0-100) ");
int grade;
scanf("%d", &grade);
grade /=10;
switch ( grade ) {
    case 10:
    case 9:
        printf("A\n");
        break;
    case 8:
        printf("B\n");
        break;
    case 7:
        printf("C\n");
        break;
    case 6:
        printf("D\n");
        break;
    default:
        printf("F\n");
        break;
}
```

这段代码不符合“单一出口”的原则，因为我们还没学过字符或字符串数据的处理

```
printf("请输入月份: ");  
int month;  
scanf("%d", &month);  
switch ( month ) {  
    case 1: printf("January\n"); break;  
    case 2: printf("February\n"); break;  
    case 3: printf("March\n"); break;  
    case 4: printf("April\n"); break;  
    case 5: printf("May\n"); break;  
    case 6: printf("June\n"); break;  
    case 7: printf("July\n"); break;  
    case 8: printf("August\n"); break;  
    case 9: printf("September\n"); break;  
    case 10: printf("October\n"); break;  
    case 11: printf("November\n"); break;  
    case 12: printf("December\n"); break;  
}
```

今后可以用数组来做

try

$$f(x) = \begin{cases} -1; & x < 0 \\ 0; & x = 0 \\ 2x; & x > 0 \end{cases}$$

- 这个分段函数能否用switch-case?