

判断

翁恺

做判断

计算时间差

- 输入两个时间，每个时间分别输入小时和分钟的值，然后输出两个时间之间的差，也以几小时几分表示

```
int hour1, minute1;  
int hour2, minute2;  
  
scanf("%d %d", &hour1, &minute1);  
scanf("%d %d", &hour2, &minute2);
```

如果直接分别减，会出现分钟借位的情况：1点40分和2点10分的差？

如果

- 就用分别减的方案，然后判断有没有出现借位行不行？
- 借位的表现是，分钟减的结果小于0

```
int im = minute2 - minute1;  
if ( im <0 ) {  
    ...  
}
```

```
int hour1, minute1;  
int hour2, minute2;
```

```
scanf("%d %d", &hour1, &minute1);  
scanf("%d %d", &hour2, &minute2);
```

```
int ih = hour2 - hour1;  
int im = minute2 - minute1;
```

```
if ( im < 0 ) {  
    im = 60 + im;  
    ih --;  
}
```

如果 (im 小于 0), 那么

```
printf("时间差是%d小时%d分。\\n", ih, im);
```

如果

```
if ( 条件成立 ) {
```

```
...
```

```
}
```


判断的条件

条件

- 计算两个值之间的关系，所以叫做关系运算

运算符	意义
<code>==</code>	相等
<code>!=</code>	不相等
<code>></code>	大于
<code>>=</code>	大于或等于
<code><</code>	小于
<code><=</code>	小于或等于

关系运算的结果

- 当两个值的关系符合关系运算符的预期时，关系运算的结果为整数1，否则为整数0
 - `printf("%d\n", 5==3);`
 - `printf("%d\n", 5>3);`
 - `printf("%d\n", 5<=3);`

优先级

- 所有的关系运算符的优先级比算术运算的低，但是比赋值运算的高
 - $7 \geq 3 + 4$
 - `int r = a > 0;`

优先级

- 判断是否相等的`==`和`!=`的优先级比其他的低，而连续的关系运算是从左到右进行的
 - `5 > 3 == 6 > 4`
 - `6 > 5 > 4`
 - `a == b == 6`
 - `a == b > 0`

找零计算器

找零计算

这个应该是类似超市之类用来进行找零计算的一个小程序，输入商品价格，收一百元，计算应该找零多少。

该程序有可以改进的地方，超市收入不可能一直是100，所以把收到的金额也写进程序，这样就可以进行任意计算了。

改进如下：

```
#include <stdio.h>

int main()
{
    int price=0;
    int get=0;

    printf("请输入金额（元）：");

    scanf("%d",&price);

    printf("请输入收入（元）：");

    scanf("%d",&get);

    int change=get-price;

    printf("找您%d元。\\n",change);

    return 0;
}
```

socovo 5月22日 来自课件“change.c”

+ 关注

← 回复

👍 4 | 举报 | 管理

小小修改了下，更使用了，呵呵

```
1  #include<stdio.h>
2  int main()
3  {
4      int price=0;
5      int total=0;
6      printf("应收金额（元）：");
7      scanf("%d",&price);
8      printf("实际收您金额：");
9      scanf("%d",&total);
10     int change=total-price;
11     printf("找您%d元\\n",change);
12     return 0;
13 }
```

C:\Users\Administrator\Desktop\未命名4.exe

```
应收金额（元）： 54
实际收您金额：80
找您26元
```

```
-----
Process exited after 10.1 seconds with return value
请按任意键继续. . .
```

找零计算器

- 找零计算器需要用户做两个操作：输入购买的金额，输入支付的票面，而找零计算器则根据用户的输入做出相应的动作：计算并打印找零，或告知用户余额不足以购买。
- 从计算机程序的角度看，这就是意味着程序需要读用户的两个输入，然后进行一些计算和判断，最后输出结果。

找零计算器

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
printf("应该找您: %d\n", bill - price);
```


注释

- 以两个斜杠“//”开头的语句把程序分成了三个部分：

1. 初始注释（comment）插入在程序代码中，
 2. 读用来向读者提供解释信息。它们对于程
 3. 计序的功能没有任何影响，但是往往能使
- 得程序更容易被人类读者理解。

`/* */`注释

- 延续数行的注释，要用多行注释的格式来写。多行注释由一对字符序列“`/*`”开始，而以“`*/`”结束。
- 也可以用于一行内的注释
 - `int ak=47 /* 36*/, y=9;`

比较

```
int price = 0;
int bill = 0;
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
printf("应该找您: %d\n", bill - price);
```

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
printf("应该找您: %d\n", bill - price);
```

高下立见

判断票面够不够

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
if ( bill >= price ) {
    printf("应该找您: %d\n", bill - price);
}
```

如果 (bill 大于等于 price) , 那么

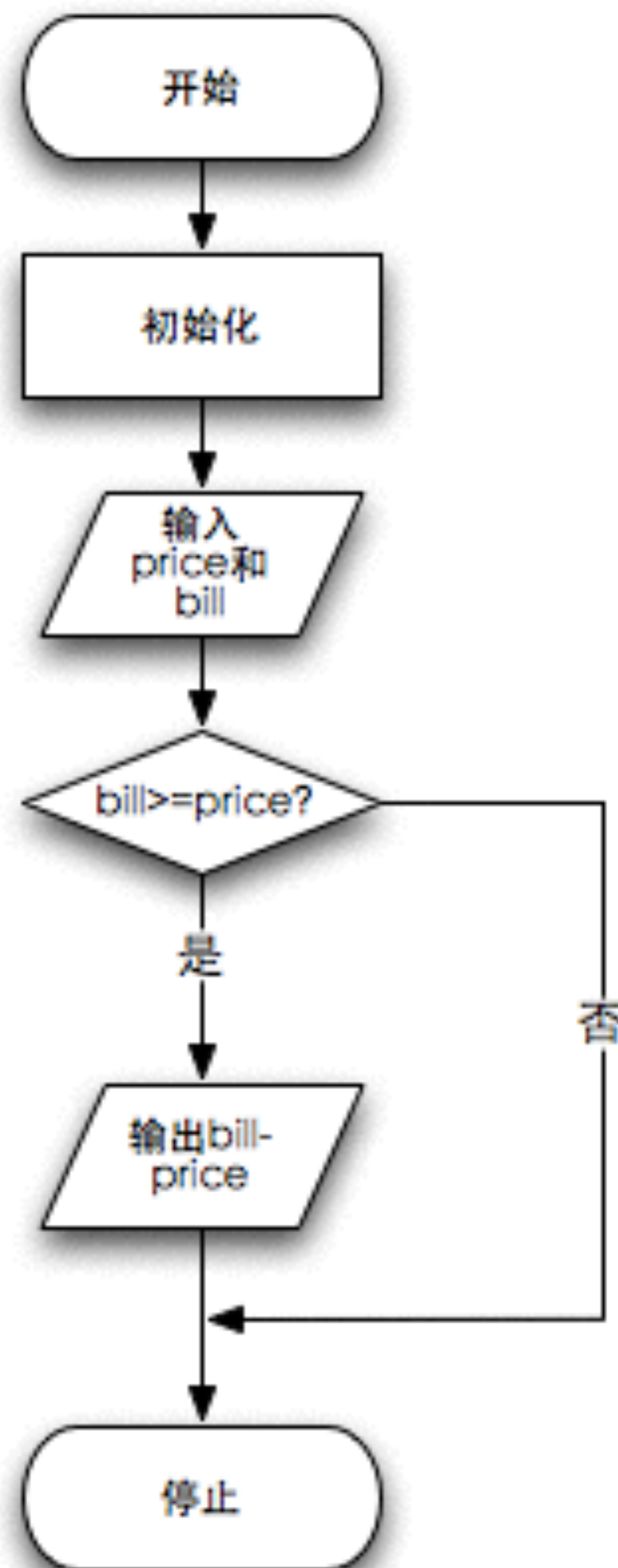
不够怎么办?

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
if ( bill >= price ) {
    printf("应该找您: %d\n", bill - price);
}
```

- 画流程图来看看

流程图

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
if ( bill >= price ) {
    printf("应该找您: %d\n", bill - price);
}
```




```
const int MINOR = 35;

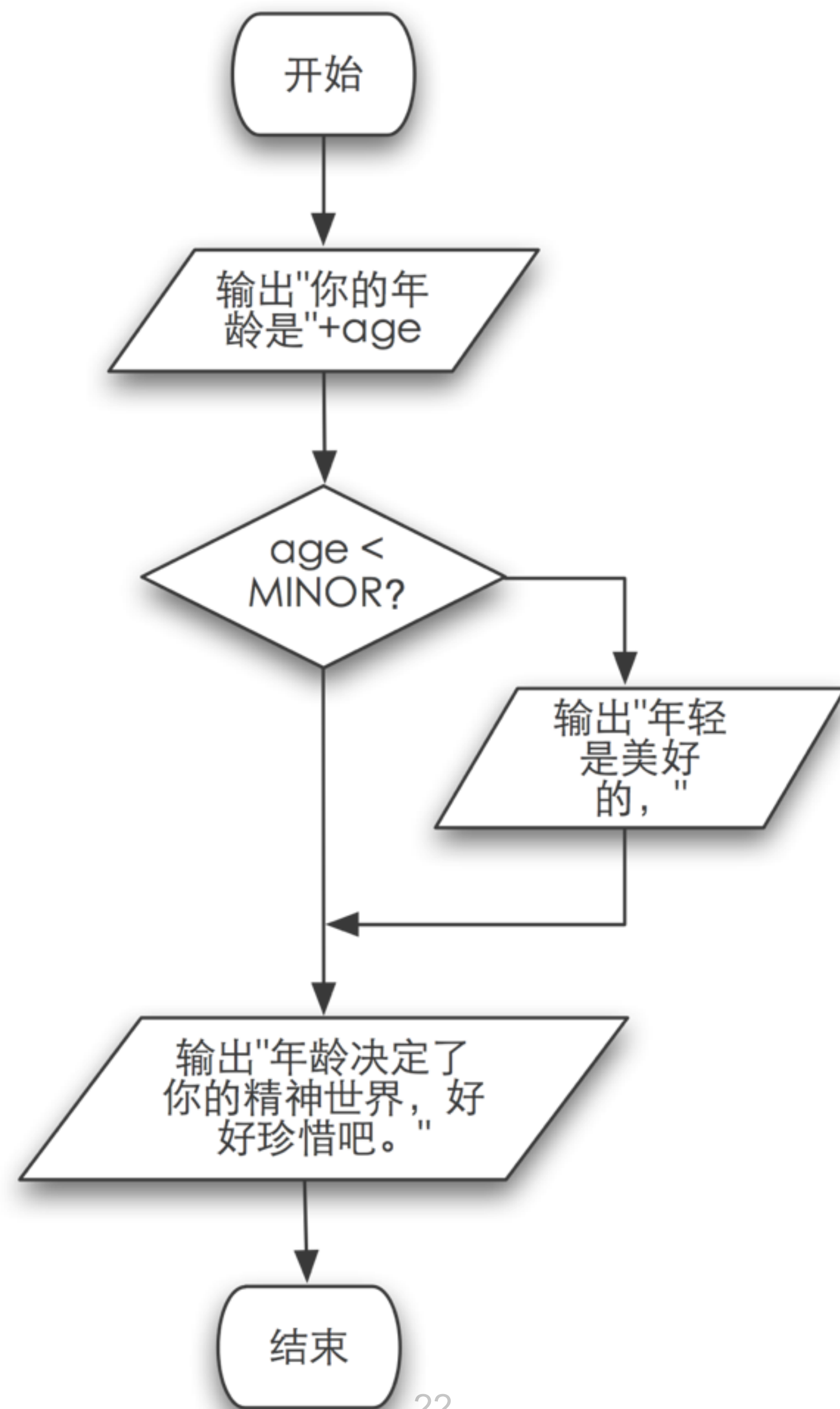
int age = 0;

printf("请输入你的年龄: ");
scanf("%d", &age);

printf("你的年龄是%d岁。\\n", age);

if ( age < MINOR ) {
    printf("年轻是美好的, ");
}

printf("年龄决定了你的精神世界, 好好珍惜吧。\\n");
```



否则的话

不够怎么办?

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
if ( bill >= price ) {
    printf("应该找您: %d\n", bill - price);
}
```

- 如果在if后面还有语句，它们在if结束后会执行，无论条件如何

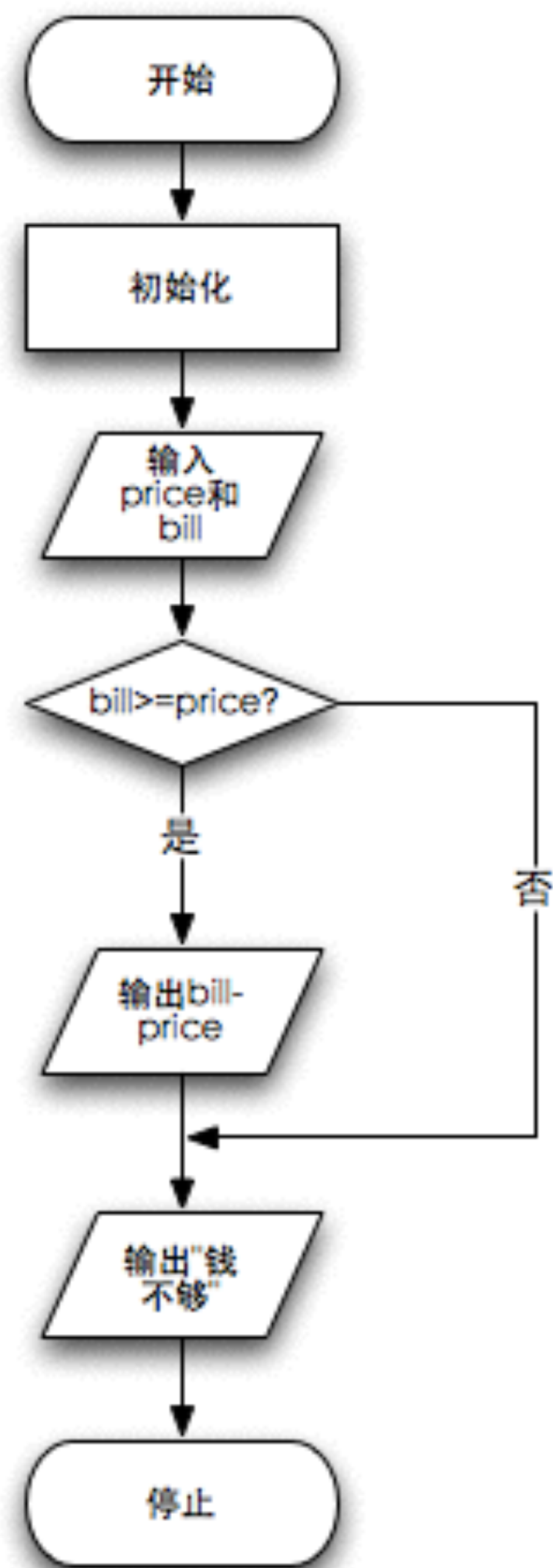
不够怎么办?

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
if ( bill >= price ) {
    printf("应该找您: %d\n", bill - price);
}
printf("你的钱不够\n");
```

- 如果在if后面还有语句，它们在if结束后会执行，无论条件如何

不够怎么办?

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
if ( bill >= price ) {
    printf("应该找您: %d\n", bill - price);
}
printf("你的钱不够\n");
```



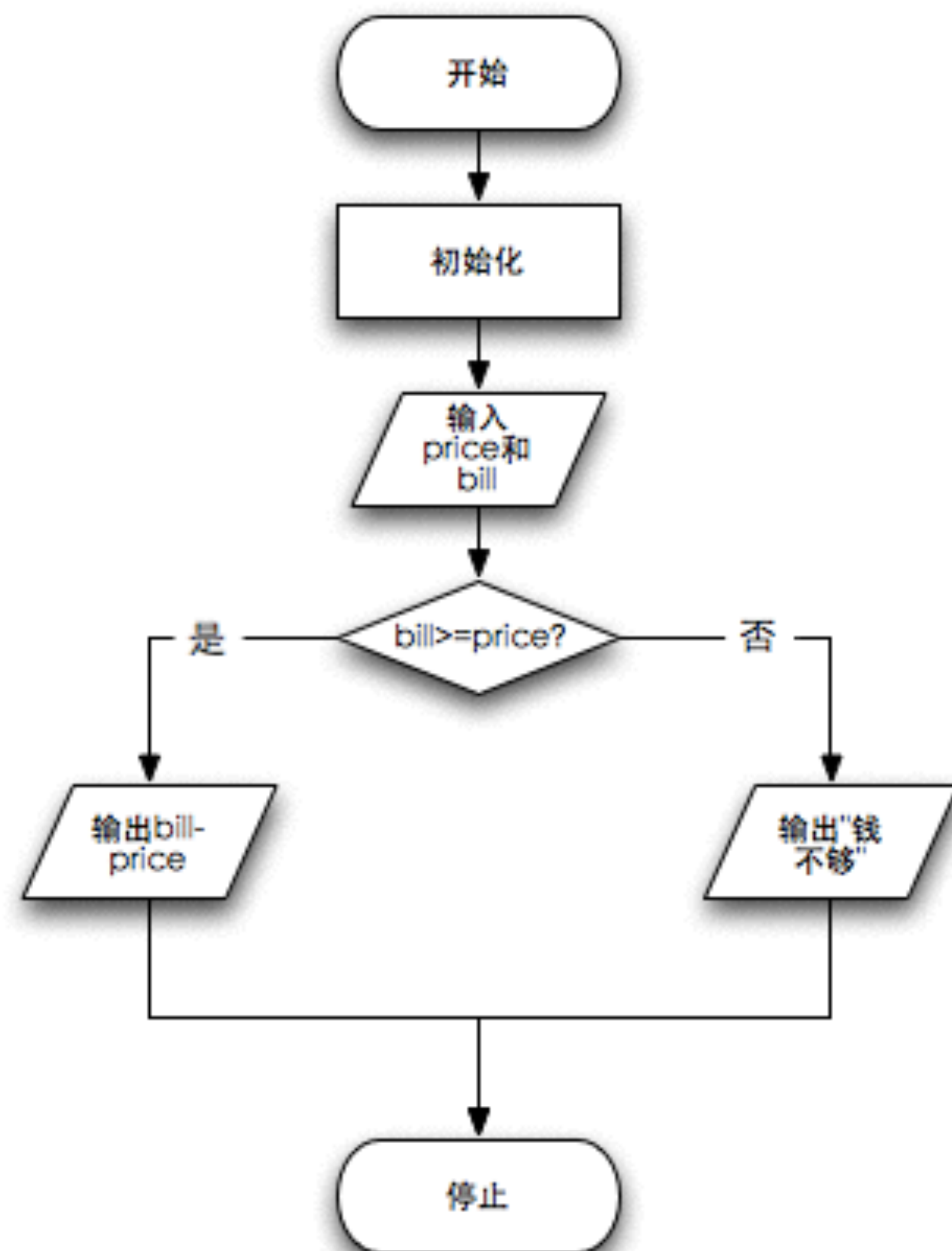
else

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
if ( bill >= price ) {
    printf("应该找您: %d\n", bill - price);
} else {
    printf("你的钱不够\n");
}
```

- else = 否则的话

else

```
// 初始化
int price = 0;
int bill = 0;
// 读入金额和票面
printf("请输入金额: ");
scanf("%d", &price);
printf("请输入票面: ");
scanf("%d", &bill);
// 计算找零
if ( bill >= price ) {
    printf("应该找您: %d\n", bill - price);
} else {
    printf("你的钱不够\n");
}
```



比较数的大小

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

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

```
printf("大的那个是%d\n", max);
```

它没有解决b大于a的问题，当a>b的条件不成立时，程序就结束了，max没有得到值

方案I

```
int a,b;

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

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

printf("大的那个是%d\n", max);
```

方案2

```
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);
```


方案3

```
int a,b;

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

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

printf("大的那个是%d\n", max);
```


* 比较一下

```
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);
```

```
int a,b;
```

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

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

```
printf("大的那个是%d\n", max);
```

if语句

if语句

- 一个基本的if语句由一个关键字if，其后是在括号里的一个表示条件的逻辑表达式，然后是一对大括号“{}”之间的语句。如果表示条件的逻辑表达式不为零，那么就执行后面跟着大括号中的语句，否则就跳过这组语句，而继续下面的其他语句。

```
if ( total > amount )
```

```
    total += amount+10;
```

if语句这一行结束的时候并没有表示语句结束的“;”，而后面的赋值语句写在if的下一行，并且缩进了，在这一行结束的时候有一个表示语句结束的“;”。这表明这条赋值语句是if语句的一部分，if语句拥有和控制这条赋值语句，决定它是否要被执行。

计算薪水

```
const double RATE = 8.25;
const int STANDARD = 40;
double pay = 0.0;
int hours;

printf("请输入工作的小时数: ");
scanf("%d", &hours);
printf("\n");
if (hours > STANDARD)
    pay = STANDARD * RATE +
        (hours-STANDARD) * (RATE * 1.5);
else
    pay = hours * RATE;
printf("应付工资: %f\n", pay);
```


判断成绩

```
const int PASS=60;  
int score;  
  
printf("请输入成绩: ");  
scanf("%d", &score);  
  
printf("你输入的成绩是%d.\n", score);  
if ( score < PASS )  
    printf("很遗憾, 这个成绩没有及格。");  
else  
    printf("祝贺你, 这个成绩及格了。");  
printf("再见\n");
```


判断成绩

```
const int PASS=60;
int score;

printf("请输入成绩: ");
scanf("%d", &score);

printf("你输入的成绩是%d.\n", score);
if ( score < PASS )
    printf("很遗憾, 这个成绩没有及格。");
else
    printf("祝贺你, 这个成绩及格了。");
printf("再见\n");
```

```
const int PASS=60;
int score;

printf("请输入成绩: ");
scanf("%d", &score);

printf("你输入的成绩是%d.\n", score);
if ( score < PASS )
    printf("很遗憾, 这个成绩没有及格。");
else {
    printf("祝贺你, 这个成绩及格了。");
    printf("再见\n");
}
```