

## 一. 创建更加友好的用户界面

1. 矛盾所在：大家都知道`getchar()`是读取单个字符，而`scanf()`是跳过所有空白的；在程序输入的过程中我们都是用Enter键来结束输入，如果这个输入队列用`scanf`读取的话，就会留下`\n`，不利于`getchar()`的读取。
2. 解决方案：在`getchar()`和`scanf()`共同使用时，我们要学会从输入队列中抛弃输入：  
`while (getchar() != '\n') {continue}` 这里做了一个小变动，就是不把`getchar()`得到的值赋给一个变量了，这样做的结果就是逐个抛弃所有`\n`之前（包括`\n`）的输入。

## 二. 输入验证

1. 输入的本质：我们的输入对C程序而言是一个字节流，这些字节里面都是相应字符的二进制编码，利用转换说明可以用`scanf`将其转化为我们想要的值。
  - ①`%d`：比如字符串' 42 '，编译器分别读取' 4 '和' 2 '，然后计算出对应的整数值，最后放在一个`int`类型的变量中。
  - ②`%f`：编译器也会读取两个字符，计算出他们对应的数值，用内部浮点发表示该值，并将结果储存在`float`类型的变量中。
  - ③`%c`和`%s`：这个比较简单，就不讲了。
2. 用户可能犯下的错误：①超过我们类型可表示的范围（比如溢出）②输入别的类型的数据。
3. 解决方案：对于①，没什么办法，只能用不等号判断是否超出范围；对于②，利用`scanf`的返回值。

## 三. 菜单浏览

这个没什么好说，利用`switch`语句即可。