

实例 1：判断 4 位回文数

所谓回文数，就是各位数字从高位到低位正序排列和从低位到高位逆序排列都是同一数值的数，例如，数字 1221 按正序和逆序排列都为 1221，因此 1221 就是一个回文数；而 1234 的各位按倒序排列是 4321，4321 与 1234 不是同一个数，因此 1234 就不是一个回文数。

本实例要求编写程序，判断输入的 4 位整数是否是回文数。

实例目标

- 掌握 if-else 语句的用法

实例分析

判断一个 4 位数是否是回文数，可以将这个 4 位数进行拆分，并进行重新组合，比较重新组合后的大小与原数大小是否相等，如果相等则该数为回文数，如果不相等，则该数不是回文数。例如，abcd 是一个 4 位数，使用 $(abcd/1000)$ 方式获取千位上的数字、使用 $(abcd/100\%10)$ 方式获取百位上的数字、使用 $(abcd/10\%10)$ 获取十位上的数字、使用 $(abcd\%10)$ 获取个位上的数字，根据回文数规则可以使用 $d*1000+c*100+b*10+a$ 表示重新组合的数字，如果 abcd 与 dbca 值相等，那么它就是一个回文数，如果 abcd 与 dbca 值不相等，那么它就不是回文数。

代码实现

```
palindrome_num = int(input("请输入一个四位数: "))
single = int(palindrome_num / 1000)
ten = int(palindrome_num / 100 % 10)
hundred = int(palindrome_num / 10 % 10)
ths = int(palindrome_num % 10)
reverse_order = ths * 1000 + hundred * 100 + ten * 10 + single
if palindrome_num == reverse_order:
    print(palindrome_num, "是回文数")
else:
    print(palindrome_num, "不是回文数")
```

首先使用 input() 函数接收输入的四位数、使用 int() 函数将接收的数据换为 int 类型并赋值给变量 palindrome_num，然后对输入的四位数进行拆分分别将个位数字、十位数字、百位数字、千位数字赋给变量 single、ten、hundred、ths，接着将获取到的数字按照回文数的特点重新组合一个新的四位数并赋值给变量 reverse_order，最后使用 if-else 语句判断变量 reverse_num 的值与变量 palindrome_num 的值是否相等，如果相等，则为回文数。

代码测试

运行代码，控制台输出结果如下（是回文数）：

请输入一个四位数：1221

1221 是回文数

运行代码，控制台输出结果如下（不是回文数）：

请输入一个四位数：1200

1200 不是回文数