

题目描述：

程序员小明打了一辆出租车去上班。出于职业敏感，他注意到这辆出租车的计费表有点问题，总是偏大。

出租车司机解释说他不喜欢数字 4，所以改装了计费表，任何数字位置遇到数字 4 就直接跳过，其余功能都正常。

比如：

- 1. 23 再多一块钱就变为 25；
- 2. 39 再多一块钱变为 50；
- 3. 399 再多一块钱变为 500；

小明识破了司机的伎俩，准备利用自己的学识打败司机的阴谋。

给出计费表的表面读数，返回实际产生的费用。

输入描述：

只有一行，数字 N，表示里程表的读数。

(1<=N<=888888888)。

输出描述：

一个数字，表示实际产生的费用。以回车结束。

补充说明：

示例 1

输入：

5

输出：

4

说明：

5 表示计费表的表面读数。

4 表示实际产生的费用其实只有 4 块钱。

示例 2

输入：

17

输出：

15

说明：

17 表示计费表的表面读数。

15 表示实际产生的费用其实只有 15 块钱。

示例 3

输入：

100

输出：

81

说明：

100 表示计费表的表面读数。

81 表示实际产生的费用其实只有 81 块钱。

```
#include <iostream>
#include <vector>
#include <cmath>
using namespace std;

int get_true(int price) {
    int true_price = price;
    for (int i = 4; i < price; ) {
        for (int j = 0; j <= i / 10; ++j) {
            int check = i / pow(10, j);
```

```

        if (check % 10 == 4) {
            true_price = true_price - pow(10, j);
            i += pow(10, min(j, 10));
            continue;
        }
    }
    i++;
}
return true_price;
}

int get_count(int a) {
    int i = 0;
    while (a / 10 > 0) {
        i++;
        a /= 10;
    }
    return i;
}

int get_true_price(int price) {
    int true_price = price;
    if (price > 4 && price < 9) {
        true_price--;
        return true_price;
    }
    int count = get_count(price);
    for (int i = 0 ; i <= count; ++i) {
        int pow_num = pow(10, i);
        int p = price % pow_num;
        if (p > 4) true_price -= pow_num * pow(10, i);
        for(int j =0; j < i && i -2 > 0; ++j){
            true_price -= 9 * pow(10, i -1);
        }
    }

    return true_price;
}

int main() {
    int price;
    while (cin >> price) { // 注意 while 处理多个 case
        cout << get_true(price);
    }
}

```

```
    }  
}  
// 64 位输出请用 printf("%lld")
```