```
/*
给你一个仅由数字 6 和 9 组成的正整数 num。

你最多只能翻转一位数字，将 6 变成 9，或者把 9 变成 6 。

请返回你可以得到的最大数字。



示例 1：

输入：num = 9669
输出：9969
解释：
改变第一位数字可以得到 6669 。
改变第二位数字可以得到 9969 。
改变第三位数字可以得到 9699 。
改变第四位数字可以得到 9666 。
其中最大的数字是 9969 。

示例 2：

输入：num = 9996
输出：9999
解释：将最后一位从 6 变到 9，其结果 9999 是最大的数。

示例 3：

输入：num = 9999
输出：9999
解释：无需改变就已经是最大的数字了。



提示：

    1 <= num <= 10^4
    num 每一位上的数字都是 6 或者 9 。

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/maximum-69-number
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

分析：

- 方法一：我们可以求出num的各位，从千位到个位遍历，遇到6就变成9，然后重新组合返回即可.
- 方法二：穷举法.

方法一:C++_遍历法

```cpp
class Solution
{
    public:
        int maximum69Number(int num)
        {
            char arr[4] = {0,0,0,0};

            arr[0] = (num/1000)     ;   // 千
            arr[1] = (num/100)%10   ;   // 百
            arr[2] = (num/10)%10    ;   // 十
            arr[3] = (num%10)       ;   // 个

            for(int i = 0 ; i < 4;i++)
            {
                if(arr[i]==6)
                {
                    arr[i]=9;
                    break;
                }
            }

            int ret_val = 0;
            for(int i = 0 ; i < 4 ; i++)
            {
                ret_val = ret_val * 10 + arr[i];
            }
            return ret_val;


        }
};


/*
执行结果:
通过
显示详情
执行用时 :4 ms, 在所有 C++ 提交中击败了63.54% 的用户
内存消耗 :8 MB, 在所有 C++ 提交中击败了100.00%的用户
*/
```

方法二：C++_穷举法

```cpp
class Solution
{
    public:
    int maximum69Number (int num)
    {
        switch(num)
        {
            case 6:
            case 9: return 9;

            case 66: return 96;
            case 69:
            case 96:
```

```
            case 99: return 99;

            case 666: return 966;
            case 669: return 969;
            case 696:
            case 966: return 996;
            case 699:
            case 969:
            case 996:
            case 999: return 999;

            case 6666: return 9666;
            case 6669: return 9669;
            case 6696: return 9696;
            case 6699: return 9699;
            case 6966:
            case 9666: return 9966;
            case 6969:
            case 9669: return 9969;
            case 6996:
            case 9696:
            case 9966: return 9996;
            case 6999:
            case 9699:
            case 9969:
            case 9996:
            case 9999: return 9999;
        }
        return 0;
    }
};

/*
执行结果：
通过
显示详情
执行用时 :4 ms, 在所有 C++ 提交中击败了63.54% 的用户
内存消耗 :8 MB, 在所有 C++ 提交中击败了100.00%的用户
*/
```

AlimyBreak
2020.01.21(2020第05周)