

Home work #0 Symmetric Numbers

1. 建立 table，儲存所有可翻轉數字與其翻轉後數字

```
string table[5][2] = {  
    {"0" , "0"},  
    {"1" , "1"},  
    {"6" , "9"},  
    {"8" , "8"},  
    {"9" , "6"},  
};
```

2. 遞迴函式，傳入幾位數 N，與目前所在 index，和之前遞迴產生的

string :front、end，數字將在每個遞迴結束時印出。

```
void symmetric_numbers(int N, int index, string front, string end);
```

3. 一般狀況(index 不為 0 且不為中間點的 index)，遞迴傳入以前的結果 front

字串再加上本次迴圈產生的新值，end 字串則加上新值的翻轉數。

```
else {  
    for (int i = 0; i < 5; i++) {  
        symmetric_numbers(N, index + 1, front + table[i][0], table[i][1] + end);  
    }  
}
```

4. 起始數字不能為 0 的例外處理，for 迴圈從 i=1 開始跑，即可排除該狀況

```
else if (!index) { //index == 0  
    for (int i = 1; i < 5; i++) {  
        symmetric_numbers(N, index + 1, front + table[i][0], table[i][1] + end);  
    }  
}
```

5. 遞迴結束條件判斷中間點(index == (int)(N / 2))，N 若為奇數，印出可放中間

的三個數:0、1、8，前後加上 front 與 end，若為偶數，直接印出 front 與

end。

```
if (index == (int)(N / 2)) {  
    if (N % 2) {  
        cout << front << '0' << end << ' ';  
        cout << front << '1' << end << ' ';  
        cout << front << '8' << end << ' ';  
    }  
    else {  
        cout << front << end << ' ';  
    }  
}
```