## 题目描述：

## 输入一个整数数组，实现一个函数来调整该数组中数字的顺序，使得所有的奇数位于数组的前半部分，所有的偶数位于数组的后半部分，并保证奇数和奇数，偶数和偶数之间的相对位置不变。

## 解题思路：

一、

## 重新定义一个vector，从前向后遍历vector，遇到奇数push\_back；再遍历vector，遇到偶数push\_back

## 二、

/\*\*

 \* 1.要想保证原有次序，则只能顺次移动或相邻交换。

 \* 2.i从左向右遍历，找到第一个偶数。

 \* 3.j从i+1开始向后找，直到找到第一个奇数。

 \* 4.将[i,...,j-1]的元素整体后移一位，最后将找到的奇数放入i位置，然后i++。

 \* 5.終止條件：j向後遍歷查找失敗。

 \*/

## 算法描述：

class Solution {

public:

void reOrderArray(vector<int> &array) {

vector<int> result;

int i;

int n = array.size();

for (i=0; i<n; i++)

{

if (array[i]%2 != 0)

result.push\_back(array[i]);

}

for (i=0; i<n; i++)

{

if (array[i]%2 == 0)

result.push\_back(array[i]);

}

array = result;

}

};

二、

class Solution {

public:

void reOrderArray(vector<int> &array)

{

if(array.empty())

return ;

int i=0,j=0;

while ( i < array.size())

{

while(i<array.size() && !iseven(array[i]))

i++;

j = i+1;

while(j<array.size() && iseven(array[j]))

j++;

if( j < array.size())

{

int val = array[j];

for(int k=j-1;k>=i;k--)

{

array[k+1]=array[k];

}

array[i++] = val;

}

else

break;

}

}

bool iseven( int n)

{

if(n%2 == 0)

return true;

return false;

}

};