## 题目描述：

## 输入一个链表，按链表从尾到头的顺序返回一个ArrayList。

## 解题思路：

第一种：

通过vector<int> value定义一个一维数组，首地址为value，通过遍历链表将链表内的元素保存到value数组中，#include <vector> 头文件中包含一个函数value.push\_back(int)，可以将一个值保存到value数组中,然后在翻转数组，将首地址value返回即可。

第二种：

递归

## 算法描述：

第一种：

/\*\*

\* struct ListNode {

\* int val;

\* struct ListNode \*next;

\* ListNode(int x) :

\* val(x), next(NULL) {

\* }

\* };

\*/

class Solution {

public:

vector<int> printListFromTailToHead(ListNode\* head) {

vector<int> res;

ListNode \*p = head;

while(p != NULL)

{

res.push\_back(p->val);

p = p->next;

}

std::reverse(res.begin(),res.end());//逆转函数

return res;

}

};

第二种：

class Solution {

public:

vector<int> value; // 必须放到外边，不能放到里边

vector<int> printListFromTailToHead(ListNode\* head) {

ListNode \*p;

p = head;

if (p != NULL)

{

if (p->next != NULL)

{

printListFromTailToHead(p->next);

}

value.push\_back(p->val);

}

return value;

}

};