



链滴

# PAT 甲级刷题实录——1019

作者: [aopstudio](#)

原文链接: <https://ld246.com/article/1580700908145>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 原题链接

<https://pintia.cn/problem-sets/994805342720868352/problems/994805487143337984>

## 思路

这题提出了一个术语——Palindromic Number，学术翻译叫做回文数，用通俗的话讲就是对称数，着写和倒着写都一样，比如最近在讨论的 20200202。本题将对称数的范围从十进制扩展到了任意进制，即在给定进制下正写和倒写都一样，比如 27 在二进制下是 11011，那么 27 就是二进制下的对称。题目给出十进制的数值，同时给出进制数，需要计算十进制数值在给定进制下是否为对称数。

经过之前的题目，这题应该非常简单。我们已经知道了十进制转其他进制的方法，用 vector 存储转出来的每一位数，最后从 vector 的首尾数字分别开始进行比较即可。需要注意的是比较位数从首尾始的下标范围，以及最后输出时最后一位后面不能带空格。代码如下：

## 代码

```
#include <iostream>
#include <vector>
using namespace std;
void print(vector<int> baseNum);
int main()
{
    int decNum, base;
    bool flag = true; //判断结果
    vector<int> baseNum; //存储转换进制后的数
    cin >> decNum >> base;
    while (decNum / base != 0) //转换进制
    {
        baseNum.push_back(decNum%base);
        decNum /= base;
    }
    baseNum.push_back(decNum);
    for (int i = 0; i < baseNum.size()/2; i++) //比较首尾数字是否相等
    {
        if (baseNum[i] != baseNum[baseNum.size() - 1 - i]) //如果存在不相等的
        {
            flag = false; //置flag为false
            break;
        }
    }
    if (flag == false)
    {
        cout << "No" << endl;
        print(baseNum);
    }
    else
    {
        cout << "Yes" << endl;
        print(baseNum);
    }
    return 0;
}
```

```
}  
void print(vector<int> baseNum)  
{  
    for (int i = baseNum.size() - 1; i >= 0; i--)  
    {  
        if (i == 0)  
            cout << baseNum[i];  
        else  
            cout << baseNum[i] << ' ';  
    }  
}
```