

九进制，回文数，短除法

九进制回文数

提示信息

回文数: 反向排列与原来一样的数就叫做回文数

例如, 12321 是回文数, 1231 不是回文数。

九进制数: 指逢9进位的一种进位计数制, 以0、1、2、3、4、5、6、7、8共九个数码表示。例如: 十进制数 9等于九进制数 10, 十进制数 81 等于九进制数 100, 以此类推。

时间限制:1000MS

内存限制:65536KB

题目描述:

给定两个正整数N和M (1<N<M<5000), 请计算出N到M之间 (含N和M) 总共有多少个数满足以下条件:

- 1、转换为九进制之后为回文数:
- 2、转换为九进制后每个数位上的数字都是奇数

例如:

当N=90,M=120,90到 120之间(含90和120) 总共有2 个数满足条件,为91 和109;

91 转换为九进制为 111, 是一个回文数, 且每个数位上的数字都是奇数;

109 转换为九进制为 131, 是一个回文数, 且每个数位上的数字都是奇数;

故输出2。

输入描述:

输入两个正整数N 和 M (1<N<M<5000), 两个正整数之间用一个空格隔开

输出描述

输出一个整数, 表示 N到M之间 (含 N和 M) 总共有多少个满足条件的正整数

样例输入

90 120

样例输出

2

```

#include<iostream>
const int N = 100;
int arr[N];
using namespace std;
// 短除法将十进制转r进制
bool isCircle(int num)
{
    for(int i = 0; i < N ;++i) arr[i] = 0; // 将数组置0
    int i =0,l;
    do{
        l = num % 9;
        arr[i] = l;
        i++;
        num /= 9;
    }while(num != 0); // 0~ i-1
    // for(int j = i-1; j>=0; j--) cout << arr[j] << ' '; // i-1表示下标从当前位置减1
    开始
    for (int j = 0; j <= (i-1)/2; j++)
    {
        if(arr[j] == arr[i-1-j]&&arr[j]%2==1&&arr[i-1-j]%2==1);
        else return false;
        /* code */
    }
    return true;
}

int main()
{
    int m, n;
    cin >> n >>m;
    int res = 0;
    for(int i = n; i < m;++i)
    {
        if(isCircle(i))
        {
            cout << i;
            res++;
        }
    }
    cout << res;

    return 0;
}
// 得到回文数
// 判断一个使用数组存储起来的数字是否为回文数，（之前使用的是重组数字的方法，构建一个数字看是否跟原来的相同）
// 判断奇数

```

判断回文数(根据数字的存储方式有三种做法)

1. 假如这个数字是用int类型表示的 int num = 12821;
2. 假如这个数字是用数组存储的 int arr[5] = {1,2,8,2,1};
3. 假如这个数字使用string存储的 string num = "128321";

```
#include<iostream>
using namespace std;
const int N = 100;
int arr[N];
bool isPali(int number)
{
    int n=0, l,x=1;
    int num = number; // 记得使用另外一个变量存储number
    while(num != 0) // 逆序构建一个新的数字
    {
        l = num%10;
        n = n + l* x;
        x *=10;
        num /= 10;
    }
    return n == number;
}
bool isPali(int arr[],int len)
{
    for (int i = 0; i < len/2; i++)
    {
        if(arr[i] != arr[len-1-i]) return false;
    }
    return true;
}

bool isPali(string num)
{
    int len = num.size();
    for (int i =0; i<len/2;i++)
    {
        if(num[i]!=num[len-1-i]) return false;
    }
    return true;
}

int main()
{
    int num;
    // string num;
    cin >> num;
    cout << isPali(num);
    return 0;
}
```

短除法进行进制转换

```
#include<iostream>
const int N = 100;
int arr[N];
int main()
{
    int m, n;
    cin >> n >> m;
    int i = 0, l;
    do{
        l = num % 9;
        arr[i] = l;
        i++;
        num /= 9;
    }while(num != 0); // 0~ i-1
    for(int j = i-1; j>=0; j--) cout << arr[j] << ' '; // i-1表示下标从当前位置减1开始
    return 0;
}
```