

将一个长度超过100位数字的十进制非负整数转换为二进制数(大数据处理)

题目描述：

将一个长度超过100位数字的十进制非负整数转换为二进制数输出。

输入：

多组数据，每行为一个长度不超过30位的十进制非负整数。
(注意是10进制数字的个数可能有30个，而非30bits的整数)

输出：

每行输出对应的二进制数。

样例输入：

```
0
1
3
8
```

样例输出：

```
0
1
11
1000
```

分析：这个数不应该存储到一个int类型变量里面去，同样和前几篇帖子一样，也是用向量来处理，处理出的结果用binary存储，最后输出就是我们要的结果

C++代码如下：

```
// w397090770
// wyphao.2007@163.com
// 2012.07.14
#include <iostream>
#include <string>
#include <vector>
```

```
using namespace std;

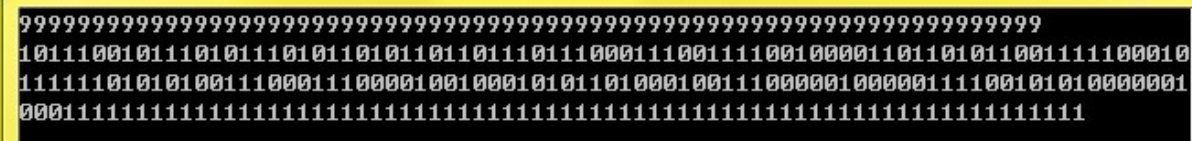
//convert decimal to Binary
void dec2bin(string s){
    int sum(0);
    vector<char>v;
    int i, j;
    string binary;
    char ch;

    //把每一位存到v里面去
    for(int i(s.length() - 1); i >= 0; --i){
        v.insert(v.begin(), s[i] - '0');
    }

    //计算二进制
    while(1){
        j = v.size() - 1;
        ch = v[j] % 2 + '0';
        binary.insert(binary.begin(), 1, ch);
        for(sum = 0, i = 0; i <= j; i++){
            if(i < j){
                //v[i + 1] += (v[i] % 2) * 10;
                v[i + 1] += (v[i] - ((v[i] >> 1) << 1)) * 10;
            }
            //v[i] /= 2;
            v[i] >>= 1;
            sum += v[i];
        }
        if(sum == 0){
            break;
        }
    }
    cout<< binary <<endl;
}

int main(){
    string dec;
    while(cin>>dec){
        dec2bin(dec);
    }
    return 0;
}
```

运行结果：

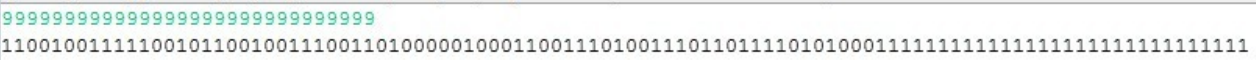


我在网上搜索了一下，看到有人用java写了一个版本，函数很简单：如下：

```
import java.math.BigInteger;
import java.util.Scanner;

public class Q1138 {
    public static void main(String[] args) {
        Scanner cin = new Scanner(System.in);
        while(cin.hasNext()){
            System.out.println(new BigInteger(cin.next()).toString(2));
        }
    }
}
```

结果：



从上面看出，java语言的确很方便，很强大，但是java代码运行时间明显要比C++实现的要长。

(转载请注明：[/archives/141](#)，请不要用于商业目的。)

本博客文章除特别声明，全部都是原创！
转载本文请加上：转载自过往记忆 (<https://www.iteblog.com/>)
本文链接：[【】 \(\)](#)