

1208: 2的幂次方表示

题目描述

任何一个正整数都可以用 2 的幂次方表示。例如 $137=2^7+2^3+2^0$ 。
同时约定方次用括号来表示，即 a^b 可表示为 $a(b)$ 。

由此可知，137 可表示为 $2(7)+2(3)+2(0)$

进一步：

$7=2^2+2+2^0$ (2^1 用 2 表示)，并且 $3=2+2^0$ 。

所以最后 137 可表示为 $2(2(2)+2+2(0))+2(2+2(0))+2(0)$ 。

又如 $1315=2^{10}+2^8+2^5+2+1$

所以 1315 最后可表示为 $2(2(2+2(0))+2)+2(2(2+2(0)))+2(2(2)+2(0))+2+2(0)$ 。

输入格式

一行一个正整数 n

输出格式

符合约定的 n 的 0, 2 表示（在表示中不能有空格）。

输入样例

1315

输出样例

$2(2(2+2(0))+2)+2(2(2+2(0)))+2(2(2)+2(0))+2+2(0)$

解析

读懂题以后，我们明白，本题的目标是将全部的数字最终都转化为由 $2(0)$ 、2 以及 $2(2)$ 进行表示。

那么，我们只需要讲 2 从 0 次方至 15 次方的值进行存储，然后再对给定的 n 进行切割计算。然后再对得到的幂值进行处理。

为什么是 15 次方呢？因为最大的数值为 20000，而最接近这个值的就是 2 的 15 次方。

编码

```
#include <bits/stdc++.h>
```

```

using namespace std;

//最大数据为20000，2的15次方32767
const int MaxNum = 15;

int a[MaxNum];

//记录2的0次方到2的15次方以内的全部数值
void initData() {
    a[0] = 1;
    for (int i = 1; i <= MaxNum; i++) {
        a[i] = a[i - 1] * 2;
    }
}

//进行递归拆分
void count(int n) {
    //找到最接近当前值的2的幂次方
    int num;
    for (int i = 0; i <= MaxNum; i++) {
        if (a[i] > n) {
            num = i - 1;
            break;
        }
    }
    cout << "2";
    //根据不同的幂数，进行不同的显示
    switch (num) {
        case 0:
            cout << "(0)";
            break;
        case 1:
            break;
        case 2:
            cout << "(2)";
            break;
        //超过2的部分继续递归
        default:
            cout << "(";
            count(num);
            cout << ")";
            break;
    }
    //对剩余的部分继续进行递归处理
    int value = n - a[num];
    if (value > 0) {
        cout << "+";
        count(value);
    }
}

```

```
}
```

```
int main(int argc, char **argv) {  
    int n;  
    initData();  
    cin >> n;  
    count(n);  
    return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

