HOMEWORK REPORT

个人信息

姓名	学号
吴宇杰	19215028

题目内容

> 给定一个整数，输出全部整数分割成其他整数的方案

解题思路

> 利用递归的思路将原问题分解，
> 用num代表需要分割的数字，max代表最大分割的数，利用循环每个数字对num分割的可能
> 若num<=max，则产生一种用num分割num的方案，然后循环比num小的数字寻求其他分割方案
> 若num>max，则循环小于等于max的寻求其他分割方案

实现代码

```
#include <iostream>
#include <vector>

using namespace std;

vector<vector<int>> res;

void num_split(int n, int u, vector<int> cur_list)
{
    if (n == 1) {
        cur_list.push_back(1);
        res.push_back(cur_list);
        return;
    }
    if (u == 1) {
        for (int i=0;i<n;i++){
            cur_list.push_back(1);
        }
        res.push_back(cur_list);
        return;
    }
    if (u > n) {
        num_split(n, n, cur_list);
```

```
        return;
    };
    if (n == u) {
        vector<int> next_list(cur_list);
        cur_list.push_back(n);
        res.push_back(cur_list);
        num_split(n, n-1, next_list);
        return;
    }
    num_split(n, u-1, cur_list);
    cur_list.push_back(u);
    num_split(n-u, u, cur_list);
}

int main() {
    int rangeNum;
    cin >> rangeNum;

    num_split(rangeNum, rangeNum, {});

    for(auto &list:res) {
        for(auto &x:list) {
            &x != &(*list.rbegin()) ?
                cout << x << "+" : cout << x;
        }
        cout << endl;
    }
}
```

测试样例

```
输入: 5
输出:
5
1+1+1+1+1
2+1+1+1
2+2+1
3+2
3+1+1
4+1
```

总结

> 对于该问题，递归关系可以表示为

$$ q(num, max)=\begin{cases} 1,\quad num = 1\ num, \quad max = 1\ q(num, num), \quad max>num \ 1 + q(num, max-1). \quad num == max \ q(num, max - 1) + q(num-max, max), \quad num < max \ \end{cases} $$

> 递归关系过于复杂，从递归关系上可以看出，每个输出代表着一种情况，没有重复利用子问题出现．
> 所以输出的个数可以近似看为其时间复杂度，输出的总数量可以用生成函数法算出

> 因为分割整数也是求组合形式，所以可以定义`P(n)`为

$$ 1取了多少次->(1+x^1+x^2+x^3+…) \ 2取了多少次->(1+x^2+x^4+x^6+…) \ 3取了多少次->(1+x^3+x^6+x^9+…) \ \vdots \ \sum_{n=0}^{\infty}p(n)x^n = (1+x^1+x^2+x^3+…)(1+x^2+x^4+x^6+…)(1+x^3+x^6+x^9+…) \cdots $$

> 上式是因为当`|x|<1`时，根据无穷级数的求解可知

$$ \sum_{i=0}^{\infty}x^i = \frac{1}{1-x} $$

> 所以可以得到`P(n)`为

$$ \sum_{n=0}^{\infty}p(n)x^n = \prod_{k=1}^{\infty}(\frac{1}{1-x^k}) $$

> 可以利用欧拉五边形数定理求解，求解过程没弄懂，略，结论是指数级的，也即时间复杂度为`O(2^n)`