

## P2089 烤鸡

## 题目描述

猪猪 Hanke 特别喜欢吃烤鸡（本是同畜牲，相煎何太急！）Hanke 吃鸡很特别，为什么特别呢？因为他有 101010 种配料（芥末、孜然等），每种配料可以放 111 到 333 克，任意烤鸡的美味程度为所有配料质量之和。

现在，Hanke 想要知道，如果给你一个美味程度 nnn，请输出这 101010 种配料的所有搭配方案。

## 输入数据

一个正整数n，表示美味程度。

## 输出数据

第一行，方案总数。

第二行至结束，101010 个数，表示每种配料所放的质量，按字典序排列。

如果没有符合要求的方法，就只要在第一行输出一个 000。

## 输入样例

11

## 输出样例

10

```
1 1 1 1 1 1 1 1 1 2
1 1 1 1 1 1 1 1 2 1
1 1 1 1 1 1 1 2 1 1
1 1 1 1 1 1 2 1 1 1
1 1 1 1 1 2 1 1 1 1
1 1 1 1 2 1 1 1 1 1
1 1 1 2 1 1 1 1 1 1
1 1 2 1 1 1 1 1 1 1
1 2 1 1 1 1 1 1 1 1
2 1 1 1 1 1 1 1 1 1
```

## 解析

方案1：直接枚举每一种材料的用量，判断其总和是否与目标相同。

## 编码

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

using namespace std;
#define rep(i) for(int i=1;i<=3;i++)

int li[60000][10];

int main(int argc, char **argv) {
    int k, ans = 0, a, b, c, d, e, f, g, h, i, j;
    cin >> k;
    rep(a) {
        rep(b) {
            rep(c) {
                rep(d) {
                    rep(e) {
                        rep(f) {
                            rep(g) {
                                rep(h) {
                                    rep(i) {
                                        rep(j) {
                                            if (a + b + c + d + e +
                                                g + h + i + j == k) {
                                                li[ans][0] = a;
                                                li[ans][1] = b;
                                                li[ans][2] = c;
                                                li[ans][3] = d;
                                                li[ans][4] = e;
                                                li[ans][5] = f;
                                                li[ans][6] = g;
                                                li[ans][7] = h;
                                                li[ans][8] = i;
                                                li[ans][9] = j;
                                                ans += 1;
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

    }
    cout << ans << endl;
    for (int i = 0; i < ans; i++) {
        for (int j = 0; j < 10; j++) {
            printf("%d ", li[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```

## 方案2：剪枝优化

我们先来确定一下第一种调料A的取值。

它存在以下4种情况：

情况1：美味值为30时，A只有一种可能，即等于3克，可以看成3到3。

情况2：美味值为29时，A有两种可能，即2克或者3克，可以看成2到3。

情况3：美味值小于等于28且大于10时，A存在三种可能，即：1克、2克或者3，可以看成1到3

情况4：美味值等于10时，A只有一种可能，即等于1克，可以看成1到1。

所以我们能够根据美味值，来提前锁定A的取值范围，即 $\max(1, n-27)$ 到 $\min(3, n-9)$ 。

当A的值锁定的时候，我们就可以来推导B的取值，即 $\max(1, n-24-A)$ 到 $\min(3, n-8-A)$ 。

以此类推：

C =  $\max(1, n-21-A-B)$  到  $\min(3, n-7-A-B)$ ,

D =  $\max(1, n-21-A-B-C)$  到  $\min(3, n-7-A-B-C)$ ,

E =  $\max(1, n-18-A-B-C-D)$  到  $\min(3, n-6-A-B-C-D)$  .....

I =  $\max(1, n-3-A-B-C-D-E-F-G-H)$  到  $\min(3, n-1-A-B-C-D-E-F-G-H)$

J =  $\max(1, n-A-B-C-D-E-F-G-H-I)$  到  $\min(3, n-A-B-C-D-E-F-G-H-I)$

## 编码

```

#include <bits/stdc++.h>

using namespace std;
#define rep(i, a, b) for (int i = max(1,a); i <= min(3,b); i++)

int li[60000][10];

int main(int argc, char **argv) {
    int n, ans = 0, a, b, c, d, e, f, g, h, i, j;
    cin >> n;
    rep(a, n - 27, n - 9) {
        rep(b, n - 24 - a, n - 8 - a) {
            rep(c, n - 21 - a - b, n - 7 - a - b) {
                rep(d, n - 18 - a - b - c, n - 6 - a - b - c) {
                    rep(e, n - 15 - a - b - c - d,

```

}

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。





· f +



