

1226 装箱问题

题目描述

一个工厂制造的产品形状都是长方体，它们的高度都是 h ，长和宽都相等，一共有六个型号，它们的长宽分别为 $1\times 1, 2\times 2, 3\times 3, 4\times 4, 5\times 5, 6\times 6$ 。这些产品通常使用一个 $6\times 6\times h$ 的长方体包裹包装然后邮寄给客户。因为邮费很贵，所以工厂要想方设法的减小每个订单运送时的包裹数量。他们很需要有一个好的程序帮他们解决这个问题从而节省费用。现在这个程序由你来设计。

输入

输入包括几行，每一行代表一个订单。每个订单里的一行包括六个整数，中间用空格隔开，分别为 1×1 至 6×6 这六种产品的数量。输入将以6个0组成的一行结尾。

输出

除了输入的最后一行6个0以外，输入文件里每一行对应着输出文件的一行，每一行输出一个整数代表对应的订单所需的最小包裹数。

输入样例

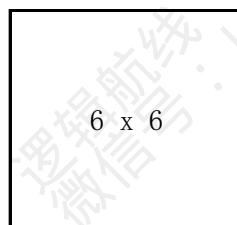
```
0 0 4 0 0 1
7 5 1 0 0 0
0 0 0 0 0 0
```

输出样例

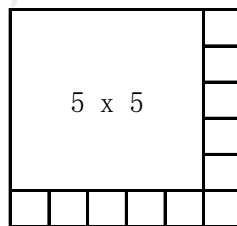
```
2
1
```

解析

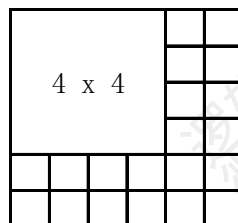
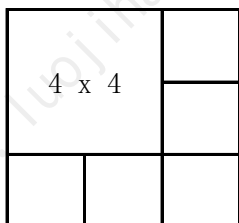
一、首先装 6×6 的，每个 6×6 的装一个箱子刚好。



二、然后装 5×5 的，每个 5×5 的需要一个箱子，箱子里多余的空间装 1×1 的，可以每个箱子装11个。

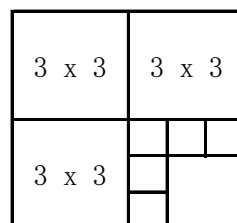
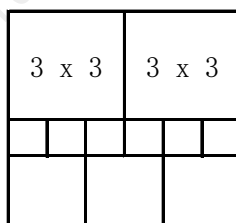
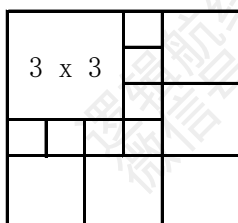
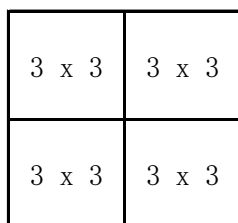


三、然后装 4×4 的，每个 4×4 的需要一个箱子。箱子里多余的空间可以装 2×2 ，也可以装 1×1 。先装 2×2 ，每个箱子最后可以装5个 2×2 ，当 2×2 不够时，再考虑装 1×1 的。每个箱子最多装20个 1×1 。



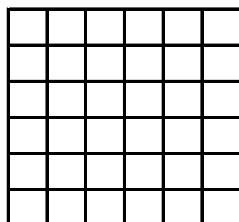
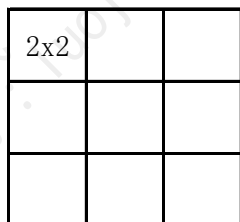
四、然后装 3×3 的。每个箱子可以装4个 3×3 的。剩下的 3×3 又分3种情形：

- 1) 剩1个 3×3 ，要装一个箱子，剩余27格空间。最多可以装5个 2×2 和7个 1×1 ；
- 2) 剩2个 3×3 ，要装一个箱子，剩余18格空间。最多可以装3个 2×2 和6个 1×1 ；
- 3) 剩3个 3×3 ，要装一个箱子，剩余9格空间，最多可以装1个 2×2 和5个 1×1 ；



然后装 2×2 的，每个箱子可以装9个 2×2 的。

最后考虑装 1×1 的，每个箱子可以装36个 1×1 ，如果还有多余的，则增加一个箱子装。



编码

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

using namespace std;
```

```

//6个型号的数量
int a, b, c, d, e, f;
//g 1*1空位数
//h 2*2空位数
int g, h;
//总共需要的箱子数
int n;

//4个3*3的产品占用1个盒子
//剩余的3*3箱子数量与有关2*2格子有如下的映射图
//剩余0个3*3,可以放入0个2*2
//剩余1个3*3,可以放入5个2*2
//.....
int k[4] = {0, 5, 3, 1};

int main() {
    //循环读入多组输入数组
    while (scanf("%d%d%d%d%d", &a, &b, &c, &d, &e, &f) != EOF) {
        //终止标记
        if (a == 0 && b == 0 && c == 0 && d == 0 && e == 0 && f == 0)
            break;
        //6*6(f)和5*5(e)和4*4(d)一定是各占一个箱子,而3*3(c) 4个占一个箱子
        n = f + e + d + ceil(c * 1.0 / 4);

        //计算2*2的空位数
        //5 * d : 一个4 * 4 必然剩余5个2*2
        //k[] : 根据剩余的3*3的块数来获取2*2 的情况
        h = 5 * d + k[c % 4];

        //实际的2*2的个数比我们留出来为2*2的空间个数多,就需要为2*2另开箱子
        if (b > h) {
            //多出来的2*2箱子应该占用的新箱子数
            //求出多需要几个2*2空间,再除以9,因为每个箱子可以放9个2*2
            n += ceil((b - h) * 1.0 / 9);
        }

        //计算剩余的1*1的空位数, n个箱子总共可以装36xn个1*1的产品
        //36 * f: 6 * 6的箱子没法装
        //25 * e: 5 * 5的占用了25个格子
        //16 * d: 4 * 4的占用了16个格子
        //9 * c: 3 * 3的占用了9个格子
        //4 * b: 2 * 2的占用了4个格子
        g = 36 * n - 36 * f - 25 * e - 16 * d - 9 * c - 4 * b;
        //如果实际1*1的个数,比我们留出的空间多就需要另开空间
        if (a > g) {
            n += ceil((a - g) * 1.0 / 36);
        }
    }
}

```

```
    }  
    printf("%d\n", 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。

