

# 矩阵乘法（分治法）

## 题目描述

设A 和 B 是两个 $n * n$ 阶矩阵，求它们的乘积矩阵C。要求使用分治法。

## 输入格式

输入为 $1+2 \times n \times n$ 个数字，每个数以空格隔开，

第1个表示矩阵阶层 $n$ ，

第2个至第 $n+1$ 个表示矩阵A，

第 $n+2$ 个至第 $2n+1$ 个表示矩阵B。

## 输出格式

输出为 $n \times n$ 个数字，表示乘积矩阵C

## 样例 #1

### 样例输入 #1

```
3 1 1 1 1 2 3 2 3 4 5 7 8 2 3 2 1 2 9
```

### 样例输出 #1

```
8 12 19 12 19 39 20 31 58
```

提示

$0 \leq n \leq 100$ , A、B均为整数矩阵

---

brute force solution

```
def solve(n, a, b):  
    # return [[sum(a[i][k] * b[k][j] for k in range(n)) for j in  
    range(n)] for i in range(n)]  
    return [sum(a[i][k] * b[k][j] for k in range(n)) for i in  
    range(n) for j in range(n)]  
  
if __name__ == '__main__':  
    n = int(input())  
    numbers = list(map(int, input().split()))  
    a = [numbers[i * n:i * n + n] for i in range(n)]  
    b = [numbers[i * n:i * n + n] for i in range(n, n + n)]  
  
    print(" ".join(map(str, solve(n, a, b))))
```

## Strassen solution

// 尚未实现