

约树

时间限制: 1.0s 内存限制: 512.0MB Special Judge

输入文件名: **divisors.in** 输出文件名: **divisors.out**

题目描述

给定正整数 n ($n \geq 3$)，你需要构造一棵 n 个节点的树，点有点权，使之满足如下要求。

- 对于任一节点，其点权为不超过 11000 的正整数。
- 对于任一不大于 n 的正整数 k ，都存在树上的一条**长度不小于 1** 的链，使得该链上所有点权的最大公约数为 k ，**其中链的长度定义为其经过的边数**。

在本题的数据范围内答案一定存在，如果有多种构造方案，输出任意一组即可。

输入格式

从 `divisors.in` 中读入数据。

输入一行一个正整数 n 。

输出格式

输出到文件 `divisors.out` 中。

输出共 n 行。

第一行输出 n 个正整数，其中第 i 个数表示你构造的树上节点 i 的点权。

接下来 $n - 1$ 行，每行包含两个正整数 u 和 v ，表示你构造的树上有一条连接 u 和 v 的边。

输出任意一组构造方案即可。

样例 1 输入

```
3
```

样例 1 输出

2 6 3
1 2
2 3

样例 1 解释

最大公约数	链的一端	链的另一端
1	1	3
2	1	2
3	2	3

样例 2 输入

4

样例 2 输出

3 6 8 12
1 2
2 3
3 4

样例 2 解释

最大公约数	链的一端	链的另一端
1	1	3
2	2	3
3	1	2
4	3	4

样例 3 输入

5

样例 3 输出

```
10 15 6 8 4
1 2
2 3
3 4
4 5
```

数据规模和约定

共 25 个测试点，每个测试点 4 分。

对于所有数据，保证 $3 \leq n \leq 2500$ 。

测试点编号	n	测试点编号	n
1 ~ 2	≤ 10	14 ~ 16	≤ 800
3	$= 32$	17 ~ 18	≤ 1200
4	$= 64$	19 ~ 21	≤ 1600
5 ~ 7	≤ 100	22 ~ 23	≤ 2000
8 ~ 10	≤ 200	24 ~ 25	≤ 2500
11 ~ 13	≤ 400		

提示

下发文件中的 `checker.cpp` 可以用于检验你的构造方案是否正确，编译指令为 `g++ checker.cpp -o checker`，执行指令为 `./checker 输入文件名 输出文件名`。

如果你的输入输出格式不符合题目要求，发生什么都是有可能的。

如果输入的 n 不在本题的数据范围内，`checker` 会输出 `Illegal input!`。

如果你构造的 w_i 不为 11000 以内的正整数，或者你构造的边连接的点的编号不为 1 到 n 中的正整数，`checker` 会输出 `Out of range!`。

如果你构造的边不构成树，`checker` 会输出 `Not a tree!`。

如果存在一个 1 到 n 中的正整数 x ，你的树上不存在一条长度不小于 1 的链，使得链上所有点权的最大公约数等于 x ，`checker` 会输出 `x not found!`。

如果你的答案正确，`checker` 会输出 `Correct!`。

注意，如果你的输入输出文件中有多个错误，`checker` 只会输出其中一个。

