

2019 年非专业级软件能力认证模拟

提高级（第二轮）

CCF-CSP-2019

第一试

时间：2019 年 9 月 24 日 08:30 ~ 12:00

题目名称	化学	工厂	质树
题目类型	传统型	传统型	传统型
可执行文件名	chem.exe	factory.exe	tree.exe
输入文件名	chem.in	factory.in	tree.in
输出文件名	chem.out	factory.out	tree.out
每个测试点时限	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MB	512 MB	512 MB
测试点/包数目	10	4	3
测试点是否等分	是	否	否

提交源程序文件名

对于 C++ 语言	chem.cpp	factory.cpp	tree.cpp
对于 C 语言	chem.c	factory.c	tree.c
对于 Pascal 语言	chem.pas	factory.pas	tree.pas

编译选项

对于 C++ 语言	-O2 -std=c++11 -Wl,--stack=536870912
对于 C 语言	-Wl,--stack=536870912
对于 Pascal 语言	

化学（chem）

【题目描述】

对一棵所有点度数不超过 4 的树，如果用树边表示 C 与 C 之间的单键，那么这棵树可以表示一种烷烃。

现给出一棵所有点度数不超过 4 的树，阅读下面的 IUPAC 有机物命名法简化版，输出该烷烃的 IUPAC 命名。

【1.1 烷烃】

烷烃统称 alkane，IUPAC 命名中以 ane 作为后缀表示为烷烃。

烷烃的化学通式为 C_nH_{2n+2} 。

【1.2 数字前缀】

对于碳个数的不同，使用不同的前缀来表示，如下表所示。

碳的个数	前缀
1	meth
2	eth
3	prop
4	but
5	pent
6	hex
7	hept
8	oct
9	non
10	dec
11	undec
12	dodec
13	tridec
14	tetradec
15	pentadec
16	hexadec
17	heptadec
18	octadec
19	nonadec
20	icos

对于超过 2 的相同取代基个数的不同，使用不同的前缀来表示，如下表所示。

碳的个数	前缀
2	di
3	tri
4	tetra
5	penta
6	hexa
7	hepta
8	octa
9	nona

【1.3 链烷烃】

如果烷烃的碳形成一条链，那么直接在 ane 前加上对应碳个数的前缀即可，例如 methane（甲烷），ethane（乙烷），pentane（戊烷）等。

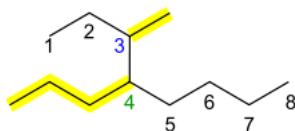
# Cs	parent name	alkane	name
1	meth		methane
2	eth		ethane
3	prop		propane
4	but		butane
5	pent		pentane

【1.4 主链】

主链是最长的一条碳链，如果有多条，取烷基（见下一条目）个数最多的一条。

主链上碳的个数决定该烷烃的大体名称，即在 ane 前加上对应碳个数的前缀。

如下图应选取标号的链作为主链，而非下方水平的这条链。

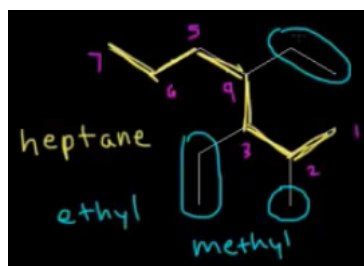


【1.5 烷基】

烷基（alkyl）是主链上取代和某个碳相连的氢原子的一个取代基，IUPAC 命名中以 y1 作为后缀表示为烷基。

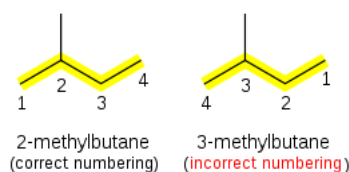
链状烷基基于碳个数，在 yl 前加上对应碳个数的前缀即为该烷基的名称，例如 methyl（甲基），ethyl（乙基），pentyl（戊基）等。

烷基如下图中蓝圈所示。



【1.6 单一烷基的命名】

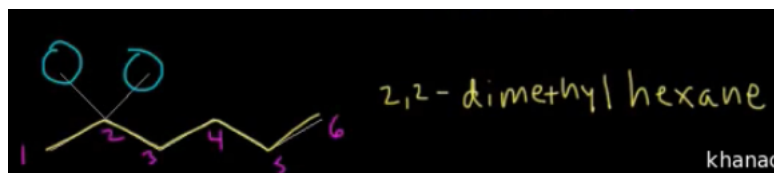
从 1 开始给主链标号，两个方向中选择使该烷基位置标号较小的标号方法，如下图所示（左图正确，右图错误）。



依次将标号，分隔符，烷基名称，主链名称连接起来即为该烷烃的命名，如上图的烷烃命名为 2-methylbutane。

【1.7 多个相同烷基的命名】

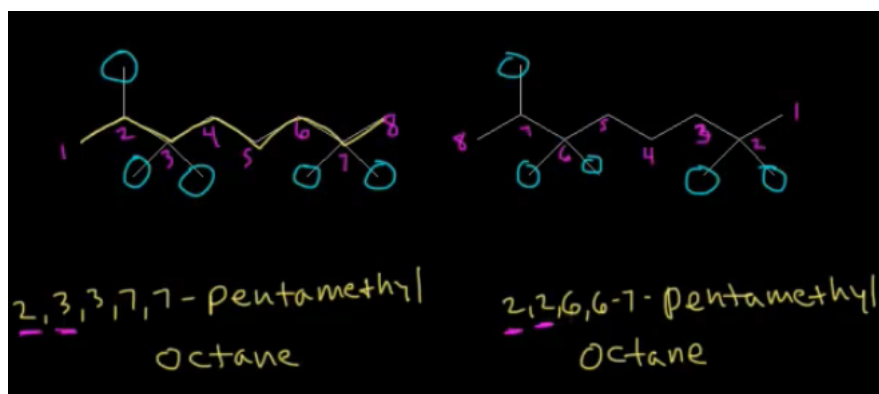
有多个相同烷基时，用逗号分隔、从小到大标示出出现的位置，并且在对应的烷基名称前加对应个数的前缀。如 2,2-dimethyl（2,2-二甲基），2,2,6,6,7-pentamethyl（2,2,6,6,7-五甲基）等。



如上图烷烃名称为 2,2-dimethylhexane。

在选取主链方向时，选取按字典序比较出现位置后较小的一种，如下图，应选取右图（2,2,6,6,7-pentamethyloctane）。

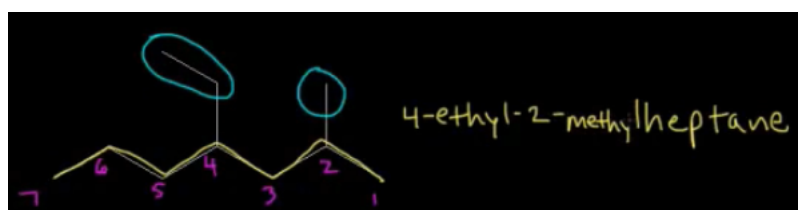
（勘误：图中 6,7 之间应用逗号而非分隔符）



【1.8 多个不同烷基的命名（每种烷基不多次出现）】

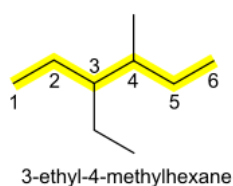
有多个不同烷基时，将不同烷基名称之间用分隔符连接起来。

选取主链时，首先选取位置标号之和较小的一种，如下图。



排列烷基之间的顺序时，按烷基的名称的字典序排列，如上图名称为 4-ethyl-2-methylheptane 而非 2-methyl-4-ethylheptane。

当两个主链方向位置标号相同但命名不同时，选取使名称里出现过的所有位置标号顺次排列后字典序尽可能小的方向。如下图，名称应为 3-ethyl-4-methylhexane 而非 4-ethyl-3-methylhexane。



【1.9 多个不同烷基的命名】

仍然用逗号分隔、从小到大标示相同烷基出现的位置，并且在对应的烷基名称前加对应个数的前缀。

注意烷基之间的字典序排序无视表示烷基个数的前缀，例如某种烷烃命名应为 3-ethyl-2,4-dimethylpentane 而非 2,4-dimethyl-3-ethylpentane。

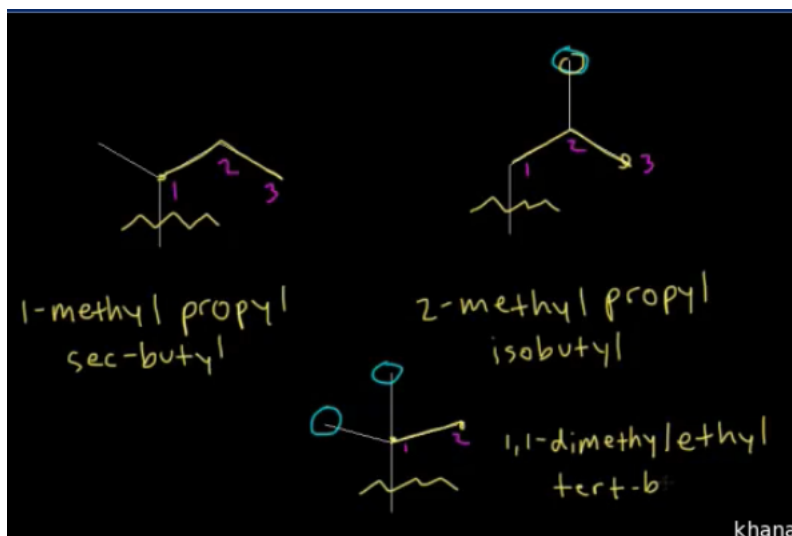
【1.10 有分支的烷基的命名】

从与主链相连的碳出发，找到最长的链作为该烷基的主链，并且以该碳为开头作为主链的方向。有多个选择烷基最多的一个。

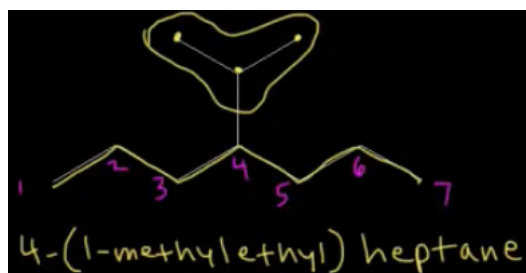
该烷基主链的命名仍为 y1 前加上对应碳个数的前缀。

该烷基的主链外的烷基的命名与前文相同。

如下图所示，锯齿线代表与主链相连的单键，加粗黄线为该烷基的主链，这三个烷基的命名分别为 1-methylpropyl, 2-methylpropyl, 1,1-dimethylethyl。



在加入有分支的烷基到主名称时需加上一对括号，如下图所示，该烷烃名称为 4-(1-methylethyl)heptane。



保证数据没有有分支的烷基的嵌套。

保证数据存在有分支的烷基时只有这一个烷基。

【输入格式】

从文件 *chem.in* 中读入数据。

输入的第一行包含一个正整数 n ，表示树的点数；

接下来 $n - 1$ 行每行两个数 u, v 描述一条树边。

【输出格式】

输出到文件 *chem.out* 中。

输出一行一个字符串表示该烷烃的 IUPAC 名称。

【样例 1 输入】

1

【样例 1 输出】

methane

【样例 2 输入】

6

1 2

2 3

2 4

4 5

4 6

【样例 2 输出】

2,3-dimethylbutane

【样例 3 输入】

10

1 2

2 3

3 4

4 5

5 6

6 7

4 8

8 9

8 10

【样例 3 输出】

4-(1-methylethyl)heptane

【提示】

若上述方法与 IUPAC 命名法有出入，请以题面为准。

保证不存在数据无法使用上述提到的方法命名。

保证不存在数据使得使用上述方法后仍有多重合法的命名。

【子任务】

当你正确实现 1.3 的内容，可获得至少 20% 的分数。

当你正确实现 1.6 的内容，可获得至少 40% 的分数。

当你正确实现 1.7 的内容，可获得至少 50% 的分数。

当你正确实现 1.8 的内容，可获得至少 60% 的分数。

当你正确实现 1.9 的内容，可获得至少 80% 的分数。

当你正确实现所有内容，可获得满分。

工厂（factory）

【题目描述】

大神 wyp 开了家工厂，工厂有 n 个工人和 p 条流水线。

工厂的工人都是睡神，因此第 i 个工人只会在 s_i 至 t_i 时刻才会工作。

每个工人都会被分派到一条流水线上，然而，一条流水线只会在这条线的工人到齐时才能开工，其余时间即使有部分工人到了也只能休息。

根据大神 wyp 的神谕，不能有流水线的工作时间为 0，也不能有工人没被分派到流水线上（即使这样会降低实际工作时间）。

工人们 dp 不过关，所以请你求出能得到的最大的工作时间总和。

保证题目至少存在一种合法的分配方案。

阅读样例以更好地理解本题。

【输入格式】

从文件 *factory.in* 中读入数据。

第一行两个正整数 n, p ;

接下来 n 行，每行两个整数表示 s_i, t_i 。

【输出格式】

输出到文件 *factory.out* 中。

一行一个整数表示答案。

【样例 1 输入】

4 2

1 3

1 5

4 6

2 7

【样例 1 输出】

4

【样例 1 解释】

最佳方案是前两个工人在 1 号流水线，后两个工人在 2 号流水线，这样每条流水线的工作时长都是 2，总时长为 4。

另一种合法方案是工人 1、2 和 4 在 1 号流水线，工人 3 在 2 号流水线，但这样总时长仅为 3。

【子任务】

本题采用子任务，当且仅当你通过一个子任务的所有测试点时才能获得该子任务的分数。

对于所有数据， $1 \leq p \leq n \leq 200$ ， $0 \leq s_i < t_i \leq 100000$ 。

子任务	分值	$n \leq$
1	20	5
2	20	20
3	20	80
4	40	200

质树（tree）

【题目描述】

大神 wyp 手里有棵二叉树，每个点有一个点权。大神 wyp 的这棵树是质树，因为随便找两个不同的点 u, v ，只要 u 是 v 的祖先，都满足 u 和 v 的点权互质。

现在你通过偷看了解到了大神 wyp 这棵树的中序遍历的点权值，你想复原出大神 wyp 的树，或者指出不可能。

阅读样例以更好地理解本题。

【输入格式】

从文件 *tree.in* 中读入数据。

第一行一个正整数 n 表示树的点数；

接下来一行 n 个正整数 a_1, a_2, \dots, a_n 表示中序遍历的点权值。

【输出格式】

输出到文件 *tree.out* 中。

如果不存在方案，输出一行 impossible；

否则输出一行 n 个数表示中序遍历的每个点的父节点在中序遍历中的位置。如果该节点是根则输出 0。

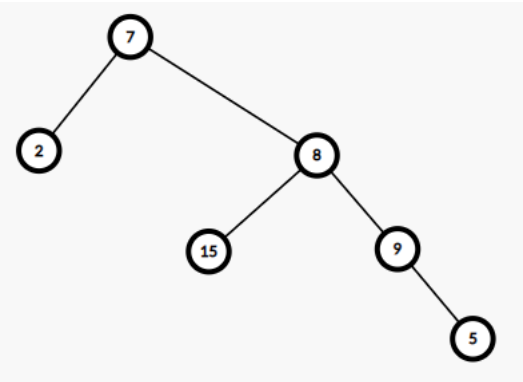
【样例 1 输入】

```
6
2 7 15 8 9 5
```

【样例 1 输出】

```
2 0 4 2 4 5
```

【样例 1 解释】



【样例 2 输入】

6
2 7 15 8 9 6

【样例 2 输出】

impossible

【子任务】

本题采用子任务，当且仅当你通过一个子任务的所有测试点时才能获得该子任务的分数。

对于所有数据， $1 \leq n \leq 10^6$ ， $1 \leq a_i \leq 10^7$ 。

子任务	分值	$n \leq$	特殊性质
1	30	1000	N/A
2	20	10^6	$a_i \leq 2$
3	50	10^6	N/A