

标题：三角形面积

已知三角形三个顶点在直角坐标系下的坐标分别为：

(2.3, 2.5)

(6.4, 3.1)

(5.1, 7.2)

求该三角形的面积。

注意，要提交的是一个小数形式表示的浮点数。

要求精确到小数后 3 位，如不足 3 位，需要补零。

标题：阅兵方阵

x 国要参加同盟阅兵活动。

主办方要求每个加盟国派出的士兵恰好能组成 2 个方阵。

x 国发现弱小的 y 国派出了 130 人的队伍，他们的士兵在行进中可以变换 2 种队形：

$$130 = 81 + 49 = 9^2 + 7^2$$

$$130 = 121 + 9 = 11^2 + 3^2$$

x 国君很受刺激，觉得 x 国面积是 y 国的 6 倍，理应变出更多队形。

于是他发号施令：

我们要派出一支队伍，在行进中要变出 12 种队形!!!

手下人可惨了，要忙着计算至少多少人才能组成 12 种不同的双方阵。

请你利用计算机的优势来计算一下，至少需要多少士兵。

(ps: 不要失去信心，1105 人就能组成 4 种队形了)

注意，需要提交的是一个整数，表示至少需要士兵数目，不要填写任何多余的内容。

标题：找假币

在 8 枚硬币中，有 1 枚假币，假币外观与真币一模一样，只是重量略轻或略重一点。

给你一架天平，要求最多称 3 次，就找出假币，并且知道它是重一些还是轻一些。

下面的代码给出一个解决方案，仔细分析逻辑，填写划线位置缺少的代码。

```
#include <stdio.h>
```

```

int balance(int a, int b)
{
    if(a<b) return -1;
    if(a>b) return 1;
    return 0;
}

void judge(char* data, int a, int b, int std)
{
    switch(balance(data[a],data[std])){
        case -1:
            printf("%d light\n", a);
            break;
        case 0:
            printf("%d heavy\n", b);
            break;
        case 1:
            printf("err!\n", b);
    }
}

// data 中 8 个元素, 有一个假币, 或轻或重
void f(char* data)
{
    switch(_____){ // 填空
        case -1:
            switch(balance(data[0]+data[4],data[3]+data[1])){
                case -1:
                    judge(data,0,3,1);
                    break;
                case 0:
                    judge(data,2,5,0);
                    break;
                case 1:
                    judge(data,1,4,0);
            }
            break;
        case 0:
            judge(data,6,7,0);
            break;
        case 1:
            switch(balance(data[0]+data[4],data[3]+data[1])){
                case -1:
                    judge(data,4,1,0);
            }
    }
}

```

```

        break;
    case 0:
        judge(data,5,2,0);
        break;
    case 1:
        judge(data,3,0,1);
    }
    break;
}
}

int main()
{
    int n;
    char buf[100];

    scanf("%d", &n);
    gets(buf);

    int i;
    for(i=0; i<n; i++){
        gets(buf);
        f(buf);
    }

    return 0;
}

```

请注意：只需要填写划线部分缺少的内容，不要抄写已有的代码或符号。

标题：约瑟夫环

n 个人的编号是 $1 \sim n$ ，如果他们依编号按顺时针排成一个圆圈，从编号是 1 的人开始顺时针报数。

(报数是从 1 报起) 当报到 k 的时候，这个人就退出游戏圈，下一个人重新从 1 开始报数。求最后剩下的人的编号。这就是著名的约瑟夫环问题。

本题目就是已知 n, k 的情况下，求最后剩下的人的编号。

题目的输入是一行，2 个空格分开的整数 n, k

要求输出一个整数，表示最后剩下的人的编号。

约定： $0 < n, k < 1$ 百万

例如输入:

10 3

程序应该输出:

4

资源约定:

峰值内存消耗 (含虚拟机) $< 256\text{M}$

CPU 消耗 $< 1000\text{ms}$

请严格按照要求输出, 不要画蛇添足地打印类似: “请输入...” 的多余内容.

注意:

main 函数需要返回 0;

只使用 ANSI C/ANSI C++ 标准;

不要调用依赖于编译环境或操作系统的特殊函数.

所有依赖的函数必须明确地在源文件中 `#include <xxx>`

不能通过工程设置而省略常用头文件.

提交程序时, 注意选择所期望的语言类型和编译器类型.

标题: 自描述序列

小明在研究一个序列, 叫 Golomb 自描述序列, 不妨将其记作 $G(n)$. 这个序列有 2 个很有趣的性质:

1. 对于任意正整数 n , n 在整个序列中恰好出现 $G(n)$ 次.
2. 这个序列是不下降的.

以下是 $G(n)$ 的前几项:

n	1	2	3	4	5	6	7	8	9	10	11	12	13
$G(n)$	1	2	2	3	3	4	4	4	5	5	5	6	6

给定一个整数 n , 你能帮小明算出 $G(n)$ 的值吗?

输入

一个整数 n .

对于 30% 的数据, $1 \leq n \leq 1000000$

对于 70% 的数据, $1 \leq n \leq 1000000000$

对于 100%的数据, $1 \leq n \leq 2000000000000000$

输出

一个整数 $G(n)$

【样例输入】

13

【样例输出】

6

资源约定:

峰值内存消耗 (含虚拟机) $< 256M$

CPU 消耗 $< 1000ms$

请严格按照要求输出, 不要画蛇添足地打印类似: “请您输入...” 的多余内容.

注意:

main 函数需要返回 0;

只使用 ANSI C/ANSI C++ 标准;

不要调用依赖于编译环境或操作系统的特殊函数.

所有依赖的函数必须明确地在源文件中 `#include <xxx>`

不能通过工程设置而省略常用头文件.

提交程序时, 注意选择所期望的语言类型和编译器类型.

标题: 采油

LQ 公司是世界著名的石油公司, 为世界供应优质石油.

最近, LQ 公司又在森林里发现了一大片区域的油田, 可以在这个油田中开采 n 个油井.

LQ 公司在这 n 个油井之间修建了 $n-1$ 条道路, 每条道路连接两个油井, 路径中间不会路过任何油井, 而且这些道路将所有油井连通.

建立油井的时候需要使用一台大型设备, 运输起来非常麻烦, LQ 公司准备在其中的一个油井位置建立一个空运站, 先将设备空运到空运站, 之后每次经过他们建立的道路来运输这个大型设备以建立不同的油井, 当油井建立完毕后再从空运站将大型设备运走.

为了减少运输的麻烦, 公司要求大型设备在道路上运输的总路程是最短的.

在建立油井和采油的过程中需要花费一些人力, 第 i 个油井需要花费 B_i 个人, 而一旦油井建成, 就需要 S_i 个人一直坚守在油井上进行维护.

当然, 如果一个人参与了油井的建设, 他可以直接留下来维护油井, 或者参与下一个油井的建设, 但是在维护油井的人不能再参加后续油井的建设了.

现在 LQ 公司想知道, 大型设备运输的总路径长度最短是多少? 在保证总路径长度最短的情况下, LQ 公司至少需要花费多少人力才能完成所有油井的建立与维护。

【输入格式】

输入的第一行包含一个整数 n , 表示油井的数量。油井由 1 到 n 依次标号。

第二行包含 n 个整数, 依次表示 B_1, B_2, \dots, B_n , 相邻的整数之间用一个空格分隔。

第三行包含 n 个整数, 依次表示 S_1, S_2, \dots, S_n , 相邻的整数之间用一个空格分隔。

接下来 $n-1$ 行描述油井之间的道路, 其中的第 i 行包含两个整数 a, b , 用一个空格分隔, 表示一条道路的起点为 $i+1$ 、终点为 a , 长度为 b , 道路是双向的, 设备可以从任意一端运送到另一端, 每条道路都可以经过任意多次。数据保证任意两个油井之间都可以通过道路连接。

【输出格式】

输出包含两个整数, 用一个空格分隔, 表示最优情况下大型设备需要运输的总路程, 以及在总路程最短的情况下最少需要花费的人力数量。

【样例输入】

```
2
10 20
15 15
1 8
```

【样例输出】

```
16 30
```

【样例说明】

有两种方案达到最优。

方案一: 在油井 2 建立空运站, 先建立油井 2, 再将大型设备运输到油井 1 建立油井 1, 最后将大型设备运回油井 2。

方案二: 在油井 1 建立空运站, 先将大型设备运输到油井 2 建立油井 2, 再将大型设备运送到油井 1 建立油井 1。

【样例输入】

```
6
3 10 20 7 15 9
2 6 10 4 8 7
1 9
1 2
2 5
3 4
3 7
```

【样例输出】

【数据规模和约定】

对于 20%的数据: n 不超过 10;

另外 20%的数据: 每个油井最多和两个油井之间有道路直接连接;

另外 10%的数据: 有 $n-1$ 个油井只有一条道路与其他油井连接;

对于 100%的数据: n 不超过 100000, B 、 S 、 c 均为不超过 10000 的正整数。

资源约定:

峰值内存消耗 (含虚拟机) $< 256\text{M}$

CPU 消耗 $< 1000\text{ms}$

请严格按照要求输出, 不要画蛇添足地打印类似: “请输入...” 的多余内容。

注意:

`main` 函数需要返回 0;

只使用 ANSI C/ANSI C++ 标准;

不要调用依赖于编译环境或操作系统的特殊函数。

所有依赖的函数必须明确地在源文件中 `#include <xxx>`

不能通过工程设置而省略常用头文件。

提交程序时, 注意选择所期望的语言类型和编译器类型。