**简单题1**

**题目 简单签到题1**

**题目描述**

给定平面上一点(x,y)，每次可以执行以下两种操作之一：

1. 让横坐标x加1
2. 让纵坐标y加2

求至少经过几次上述操作之后能够使得该点落在y=x直线上？

**输入格式**

第一行为测试组数t

接下来有t组数据，每组数据各占一行，每行为x和y的值

**输出格式**

输出t行答案

**数据范围**

0<t<107，-109<x<y<109

**测试样例**

**Input**

2

5 10

9 3

**Output**

5

3

**简单题2**

**题目 简单签到题2**

**题目描述**

有t个区间[l,r]，问最多能选取多少个区间使得两两互不相交（不计端点处）？

**输入格式**

第一行为t；

接下来的t行为各个区间l和r的值

**输出格式**

输出最多能够选取的区间个数

**数据范围**

0<t<107，-109<l<r<109

**测试样例**

**Input**

3

0 2

2 4

1 3

**Output**

2

**解释**

选取区间[0,2]和[2,4]，最多能够选取的区间个数为2。

**题目3**

David 作为一名 scut的大学生，自然需要去学习。

在 scut，有N门不同的课程，一个学期一共有M周，对于每一周，会上N次课，每一门课程恰好一周上一次课，David对于每一个课程都有一个熟练度，初始时都为 0。

对于一节课 David可以选择如下选项中的一个：

* 去上课：如果他上的是第i门课，那么他对于这节课的熟练度增加Ai。
* 翘课：David热爱学习，所以他会选一门课自学，如果他选了第i门课，那么他对于这节课的熟练度增加Bi。

为了去更多的学习课外知识，David不会在课下学习这N门课程，但是他又不想要让自己挂科，于是他找到了你，他想让自己对每一门课的熟练度的最小值最大。

**输入格式**

第一行两个整数N,M

接下来一行N个整数Ai。

接下来一行N个整数Bi。

**输出格式**

仅输出一行一个整数表示你的答案。

**样例**

**输入**

3 3

19 4 5 上课

2 6 2 课程不一样

**输出**

18

**解释**

举个例子，如果David按如下方式学习，则他对课程1,2,3的熟练度将分别为19,18,19。

第一周课程1的课：翘课自学课程2；

第一周课程2的课：翘课自学课程2；

第一周课程3的课：去上课程3的课；

第二周课程1的课：去上课程1的课；

第二周课程2的课：翘课自学课程3；

第二周课程3的课：去上课程3的课：

第三周课程1的课：翘课自学课程3；

第三周课程2的课：翘课自学课程2；

第三周课程3的课：去上课程3的课。。

**数据范围**

1<=N<=3\*10^5，1<=M,Ai,Bi<=10^9

提示：

1.本题主要算法为二分答案，check函数中运用到贪心的思想

2.所有数据都开long long

3.自己写一个上取整函数，double可能有丢精度问题

4.二分主要代码

#include <iostream>

#define ll long long

using namespace std;

ll a[300010],b[300010];

ll n, m;

ll ceil(ll x, ll y) {

}

bool check(ll k) {//最小为k

}

int main(){

cin >> n >> m;

for (int i = 1; i <= n; i++) cin >> a[i];

for (int i = 1; i <= n; i++) cin >> b[i];

ll left = 0, right = 1e18+1000;

while (left + 1 < right) {

ll mid = left + right >> 1;

if (check(mid)) left = mid;//课时数够，提高要求

else right = mid;

}

// [[[[[[ ))))) 满足课时数"[" 的最大值答案在left

cout << left;

return 0;

}

**题目4 摘果子**

**题目描述**

在一条路上，有 n 棵果树，从左到右编号为 1,2,…,n。小明有 h 个小时的时间，他希望利用这个时间摘到更多的果子。他从 1 出发，向右走（不可往回走），有选择的在一些果树停留一定的时间（是 5 分钟的倍数）摘果子。最后在某一个果树结束摘果子。小明从第 i 个果树到第 i+1 个果树需要走 5\*Ti 分钟，在第 i 个果树停留，第一个 5 分钟可以摘到 Fi个果子，以后每再摘 5 分钟，可以摘到的果子量减少 Di，若减少后的果子量小于 0，则减少后的果子量为 0 。小明想知道自己能获得的最大果子数量，请你帮帮他。

（提示：贪心+优先队列）

**输入格式**

第一行一个整数 n，表示果树的个数

第二行一个整数 h，表示小明的空闲时间

第三行有 n 个整数，Fi依次表示每个果树第一个 5 分钟能摘到果子的数量

第四行有 n 个整数，Di依次表示以后的每5分钟摘果子数量比前一个 5 分钟摘果子数量减少的数量

第五行有 n-1 个整数，Ti表示由第 i 个果树到第 i+1 个果树需要花 5 \*Ti分钟的路程

**输出格式**

输出只有一行，表示小明最多能摘果子的数量。

**数据范围**

对于所有数据，2<=n<100,1<=h<=20，1<=Fi<=10^9,0<=Di<=10^9。

**测试样例**

**Input**

3

1

4 5 6

1 2 1

1 2

**Output**

35

**解释**

在第 1 个果树摘 15 分钟，共摘得 4+3+2=9 个果子；

在第 2 个果树摘 10 分钟，共摘得 5+3=8个果子；

在第 3 个果树摘 20 分钟，共摘得 6+5+4+3=18 个果子；

从第 1 个果树走到第 2 个果树，从第 2 个果树走到第 3 个果树，共用时间 15 分钟。

共得 35 个果子，并且这是最多的数量。

**难题2**

**题目 摘果子**

**题目描述**

在一条路上，有 n 棵果树，从左到右编号为 1,2,…,n。小明有 h 个小时的时间，他希望利用这个时间摘到更多的果子。他从 1 出发，向右走（不可往回走），有选择的在一些果树停留一定的时间（是 5 分钟的倍数）摘果子。最后在某一个果树结束摘果子。小明从第 i 个果树到第 i+1 个果树需要走 5\*Ti 分钟，在第 i 个果树停留，第一个 5 分钟可以摘到 Fi个果子，以后每再摘 5 分钟，可以摘到的果子量减少 Di，若减少后的果子量小于 0，则减少后的果子量为 0 。小明想知道自己能获得的最大果子数量，请你帮帮他。

（提示：贪心+优先队列）

**输入格式**

第一行一个整数 n，表示果树的个数

第二行一个整数 h，表示小明的空闲时间

第三行有 n 个整数，Fi依次表示每个果树第一个 5 分钟能摘到果子的数量

第四行有 n 个整数，Di依次表示以后的每5分钟摘果子数量比前一个 5 分钟摘果子数量减少的数量

第五行有 n-1 个整数，Ti表示由第 i 个果树到第 i+1 个果树需要花 5 \*Ti分钟的路程

**输出格式**

输出只有一行，表示小明最多能摘果子的数量。

**数据范围**

对于所有数据，2<=n<100,1<=h<=20，1<=Fi<=10^9,0<=Di<=10^9。

**测试样例**

**Input**

3

1

4 5 6

1 2 1

1 2

**Output**

35

**解释**

在第 1 个果树摘 15 分钟，共摘得 4+3+2=9 个果子；

在第 2 个果树摘 10 分钟，共摘得 5+3=8个果子；

在第 3 个果树摘 20 分钟，共摘得 6+5+4+3=18 个果子；

从第 1 个果树走到第 2 个果树，从第 2 个果树走到第 3 个果树，共用时间 15 分钟。

共得 35 个果子，并且这是最多的数量。