OJ 更新 目前已經完成，如果有任何使用上的問題歡迎直接提出來(?)

✔ 123 . 神乎其技的壽司師傅小Q

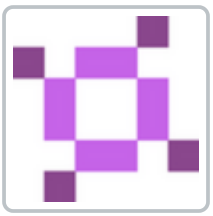
Submit (/problems/123/submissions/new)

Status (/problems/123/submissions)

Ranklist (/problems/123/ranklist)

Back to Problems List (/problems)

TopCoder



(/users/algo021)

coffee5427 (/users/algo021)

User's AC Ratio

62.5% (5/8 (/problems/123/ranklist))

Submission's AC Ratio

30.4% (7 (/problems/123/submissions?filter_status=AC)/23 (/problems/123/submissions))

Tags

enumerate (/problems/tag/enumerate) enumerate26 (/problems/tag/enumerate26)

Description

小 Q 身為一位專業的壽司師傅，能夠同時使用兩隻手各自包不同的壽司，如此特殊的才藝也替壽司店吸引了不少人潮。

某天，壽司店舉辦了一場免費吃壽司的活動，並且要請小 Q 展示一下單手包壽司的技能。已知活動當天來到壽司店的第 1 位客人將可以免費獲得 1 個壽司、第 2 位客人將可以免費獲得 2 個壽司 $\cdots$ 依此類推。但是壽司店準備的材料有限，所以小 Q 將用以下的方法捏壽司：已知當天小 Q 的左手邊準備了可以捏出 L 個壽司的材料、右手邊準備了可以捏出 R 個壽司的材料。當第 i 個客人來時，會使用所剩材料較多的那隻手捏 i 個壽司給第 i 個客人。由於小 Q 是個左撇子，因此若左手右手邊的食材一樣多時，會使用左手邊的材料捏壽司給客人。若左手邊及右手邊的材料都無法再捏出 i 個壽司時，第 i 個客人將拿不到任何壽司，並且當日的活動會立即結束。

註：左手邊及右手邊的材料不可混用，即使單邊的材料不夠也不能使用兩邊的材料組成 i 個壽司

給定小 Q 左右手邊的壽司材料數 L, R，求當天有幾個客人可以拿到壽司？並求活動結束後左邊及右邊分別剩下幾份材料？

Input Format

輸入第一行包含一個整數 T（$1 \leq T \leq 100$），接下來會有 T 組測試資料。第二行以後的 T 組測試資料中，每行各有兩個以空白分隔的整數 L, R（$1 \leq L, R \leq 10^{18}$），代表當天左右手邊準備的壽司材料數。

Output Format

輸出總共有 T 行，每行有 3 個以空白分隔的整數 n, l, r 表示當天有拿到壽司的客人數 n、當天左手邊剩下的壽司材料數 l、以及當天右手邊剩下的壽司材料數 r。

Sample Input

copy

```
// Sample input 1
3
1 2
2 2
8 11

// Sample input 2
1
1000000000000000000 1000000000000000000
```

Sample Output

copy

```
// Sample output 1
1 1 1
2 1 0
5 0 4
// Sample output 2
1999999999 0 1000000000
```

Hints

Problem Source

Google Code Jam 2020 Round 2

Solution (Click to toggle)

本題的輸入範圍可達 10^{18}，因此若只是模擬一次過程會超時。

已知發給第一個客人的壽司會從 L, R 中較大的那邊製作，且會一直持續直到兩者的大小關係改變。以下不失一般性假設 $L > R$，則前 i 個客人的壽司皆會使用較大的 L 製作，並且前 i 個客人總共會花費 $(i+1) \cdot i/2$ 個壽司材料，因此第一步可以使用二分搜找出滿足 $L - (i+1) \cdot i/2 >= R$ 的最大 $i^\star$ 值，代表前 $i^\star$ 個客人的壽司皆來自 L。

第 $i^\star + 1$ 位客人會由仍然大於等於 R 的 L 製作壽司，但製作完後的 $L - i < R$，因此會換成 R 製作第 $i + 1$ 位客人的壽司。經過此兩步驟後可以發現 $L - i > R - (i+1)$，因此又回到使用 L 製作壽司。假設 L 再第 $i^\star$ 位客人後接著執行了 x_1 步、R 再第 i^* 位客人後接著執行了 x_2 步，則 L 最後會剩下

$$L - (i^\star + 1) - (i^\star + 3) - (i^\star + 5) - \cdots - (i^\star + (2 \times (x_1) - 1)) = L - i^\star \times x_1 - x_1^2$$

因此也可以對 x_1 的值域二分搜出 x_1 可能的最大值，同理亦可以求得 x_2 為滿足

$$R - i^\star \times x_2 - (x_2 + 1) \times x_2 \geq 0$$

的最大值。

因此最後的答案為 $n = i^\star + x_1 + x_2$、$l = L - i^\star \times x_1 - x_1^2$、$r = R - i^\star \times x_2 - (x_2 + 1) \times x_2$。

Solution Code

```
#include <bits/stdc++.h>
using namespace std;

long long x;

inline long long f1(long long i) {
    return (i+1)*i/2;
}

inline long long f2(long long i) {
    return x * i + i*i;
}

inline long long f3(long long i) {
    return (x + 1) * i + i*i;
}

long long bin_search(long long thrd, long long (*func)(long long i), long long l, long lo
ng r) {
    // Return the maximum of i satisfying func(i) <= thrd
    long long mid = (l+r)>>1;
    while (r - l > 4) {
        if (func(mid) > thrd) {
            r = mid - 1;
        }
        else {
            l = mid;
        }
        mid = (l+r)>>1;
    }
    while (func(r) > thrd) {
        r--;
    }
    return r;
}

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    long long L, R, T;
    cin >> T;
    for (int t = 1; t <= T; t++) {
        cin >> L >> R;

        // Phase I
        x = bin_search(abs(L-R), f1, 0, 2000000000);
        if (L >= R) {
```

```
            L -= f1(x);
        }
        else {
            R -= f1(x);
        }

        // Phase II
        bool exch = R > L;
        if (R>L)
            swap(R, L);
        long long x1 = bin_search(L, f2, 0, 2000000000);
        long long x2 = bin_search(R, f3, 0, 2000000000);

        if (exch)
            cout << x + x1 + x2 << ' ' << R - f3(x2) << ' ' << L - f2(x1);
        else
            cout << x + x1 + x2 << ' ' << L - f2(x1) << ' ' << R - f3(x2);
        if (t != T)
            cout << endl;
    }
}```
```

Subtasks

No.	Testdata Range	Constraints	Score
1	0~1	範例測資	0
2	2~6	$1 \leq L, R \leq 1000$	30
3	0~12	無額外限制	70

Testdata and Limits

No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	524288	65536	1 3
1	1000	524288	65536	1 3
2	1000	524288	65536	2 3
3	1000	524288	65536	2 3
4	1000	524288	65536	2 3
5	1000	524288	65536	2 3
6	1000	524288	65536	2 3

No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
7	1000	524288	65536	3
8	1000	524288	65536	3
9	1000	524288	65536	3
10	1000	524288	65536	3
11	1000	524288	65536	3
12	1000	524288	65536	3

Submit (/problems/123/submissions/new)

Status (/problems/123/submissions)

Ranklist (/problems/123/ranklist)

Back to Top