

Arithmetic Number

给定一个等差数列，其中缺少了一个数，找到缺少的数

采用二分法

```
#include <bits/stdc++.h>

using namespace std;

int arithmeticNum(vector<int> &arr, int low, int high, int diff) {
    while (low <= high) {
        int mid = low + (high - low) / 2;
        if (arr[mid + 1] - arr[mid] != diff) return arr[mid] + diff;
        if (mid > 0 && arr[mid] - arr[mid - 1] != diff) return arr[mid - 1] + diff;
        if (arr[mid] == arr[0] + mid * diff)
            low = mid + 1;
        else
            high = mid - 1;
    }
    return -1;
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        scanf("%d", &N);
        vector<int> arr(N);
        for (int i = 0; i < N; ++i) scanf("%d", &arr[i]);
        printf("%d\n", arithmeticNum(arr, 0, N - 1, (arr[N - 1] - arr[0]) / N));
    }
}
```

拓展：

给定整数A为等差数列第一项，C为公差，判断数字B是否存在以A为第一项，公差为C的等差数列中

```
#include <bits/stdc++.h>

using namespace std;

bool arithmeticNum(int A, int B, int C) {
    if ((A > B && C > 0) || (B > A && C < 0)) return false;
    else if (C == 0 && A == B) return true;
    int x = ceil((B - A) / (C * 1.0));
    int y = floor((B - A) / (C * 1.0));
    return C && (x == y);
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int A, B, C;
        scanf("%d %d %d", &A, &B, &C);
        printf("%d\n", arithmeticNum(A, B, C));
    }
}
```