Find closest number in array

> 在给定数组中查找与指定target最接近的数

采用二分法，在查找过程中有如下三种情况

1. arr[mid] == target，则返回arr[mid]
2. arr[mid] > target，此时最接近target的数在arr的左边部分中，此时又细分为两种情况：
    1. mid > 0且 target > arr[mid - 1]，此时最接近target的数要么是arr[mid]，要么是arr[mid - 1]
    2. 否则令high = mid，继续查找
3. arr[mid] < target，此时最接近target的数在arr的右边部分中，此时又细分为两种情况：
    1. mid < N - 1且target < arr[mid + 1]，此时最接近target的数要么是arr[mid]，要么是arr[mid + 1]
    2. 否则令low = mid + 1，继续查找

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

using namespace std;

int compare(int &a, int &b, int &target) {
    return a - target >= target - b ? b : a;
}

int closestVal(vector<int> &arr, int low, int high, int &target) {
    if (target <= arr[low]) return arr[low];
    if (target >= arr[high]) return arr[high];
    int mid;
    while (low < high) {
        mid = low + (high - low) / 2;
        if (arr[mid] == target) return target;
        else if (target < arr[mid]) {
            // search in left part
            if (mid > 0 && target > arr[mid - 1])
                return compare(arr[mid], arr[mid - 1], target);
            high = mid;
        } else {
            // search in right part
            if (mid < arr.size() - 1 && target < arr[mid + 1])
                return compare(arr[mid + 1], arr[mid], target);
            low = mid + 1;
        }
    }
    return arr[mid];
}


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