Decrease and conquer

change programming language to c++

Insertion sort
q2.

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

void Insertion_Sort(int arr[], int n) {

    for(int i = 1; i < n; i++) {
        int j = i;

        // Swaps until the element is at its right position
        while(j > 0 && arr[j-1] > arr[j]) {
            swap(arr[j-1], arr[j]);
            j--;
        }
    }
}

int main() {
    int n;
    cin >> n;

    int arr[n];
    for(int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    Insertion_Sort(arr, n);

    for(int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << "\n";

    return 0;
}
```

q4.
Sorted - [1, 2, 3, 4, 6], Unsorted - [5]

Computing a Median and the Selection Proble

q1.

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

```
using namespace std;

double find_median(int arr[], int n) {
    double ans = 0;
    sort(arr, arr + n); // sort the array in ascending order

    if(n % 2 == 0) {
        // If the size of the array is even, return the average of the two middle elements
        ans = (arr[n/2 - 1] + arr[n/2]) / 2.0;
    } else {
        // If the size of the array is odd, return the middle element
        ans = arr[n/2];
    }

    return ans;
}

int main() {
    int t; cin >> t;
    while(t--) {
        int n; cin >> n;
        int arr[n];
        for(int i = 0; i < n; i++) {
            cin >> arr[i];
        }

        cout << setprecision(1) << fixed << find_median(arr, n) << "\n";
    }
    return 0;
}
```

q2.

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

double find_median(int arr[], int n) {
    double ans = 0;
    sort(arr, arr + n); // sort the array in ascending order

    if(n % 2 == 0) {
        // If the size of the array is even, return the average of the two middle elements
        ans = (arr[n/2 - 1] + arr[n/2]) / 2.0;
    } else {
        // If the size of the array is odd, return the middle element
        ans = arr[n/2];
    }

    return ans;
}
```

```
int main() {
    int t; cin >> t;
    while(t--) {
        int n; cin >> n;
        int arr[n];
        for(int i = 0; i < n; i++) {
            cin >> arr[i];
        }

        cout << setprecision(1) << fixed << find_median(arr, n) << "\n";
    }
    return 0;
}
```

q4.
click submit
q6.
click submit

Fake coin problem

q1.
click submit
q2.

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

char measure(int left[], int left_size, int right[], int right_size);

int solution(int n) { // n is the number of coins
    int a[1] = {1};
    int b[1] = {2};
    char c = measure(a, 1, b, 1);
    if(c == '<') {
        return 1;
    }
    else if(c == '>') {
        return 2;
    }
    int ans;
    for(int i = 3; i <= n; i++) {
        int a[1] = {1};
        int b[1] = {i};
        if(measure(a, 1, b, 1) != '=') {
            ans = i;
            break;
        }
    }
    return ans;
}
```

q3.
N
q4.
click submit
q6.
click submit
q7.
1.58