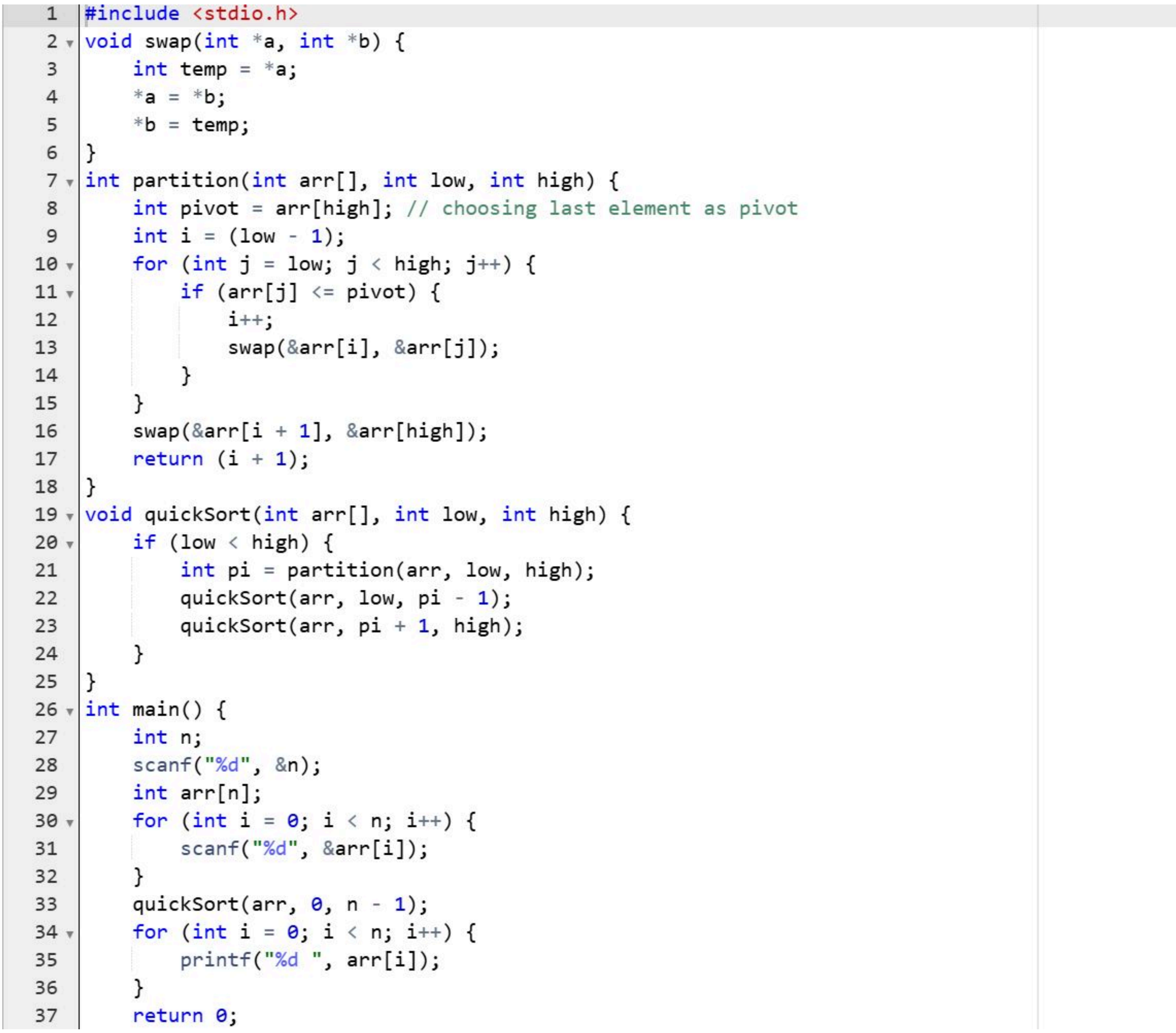
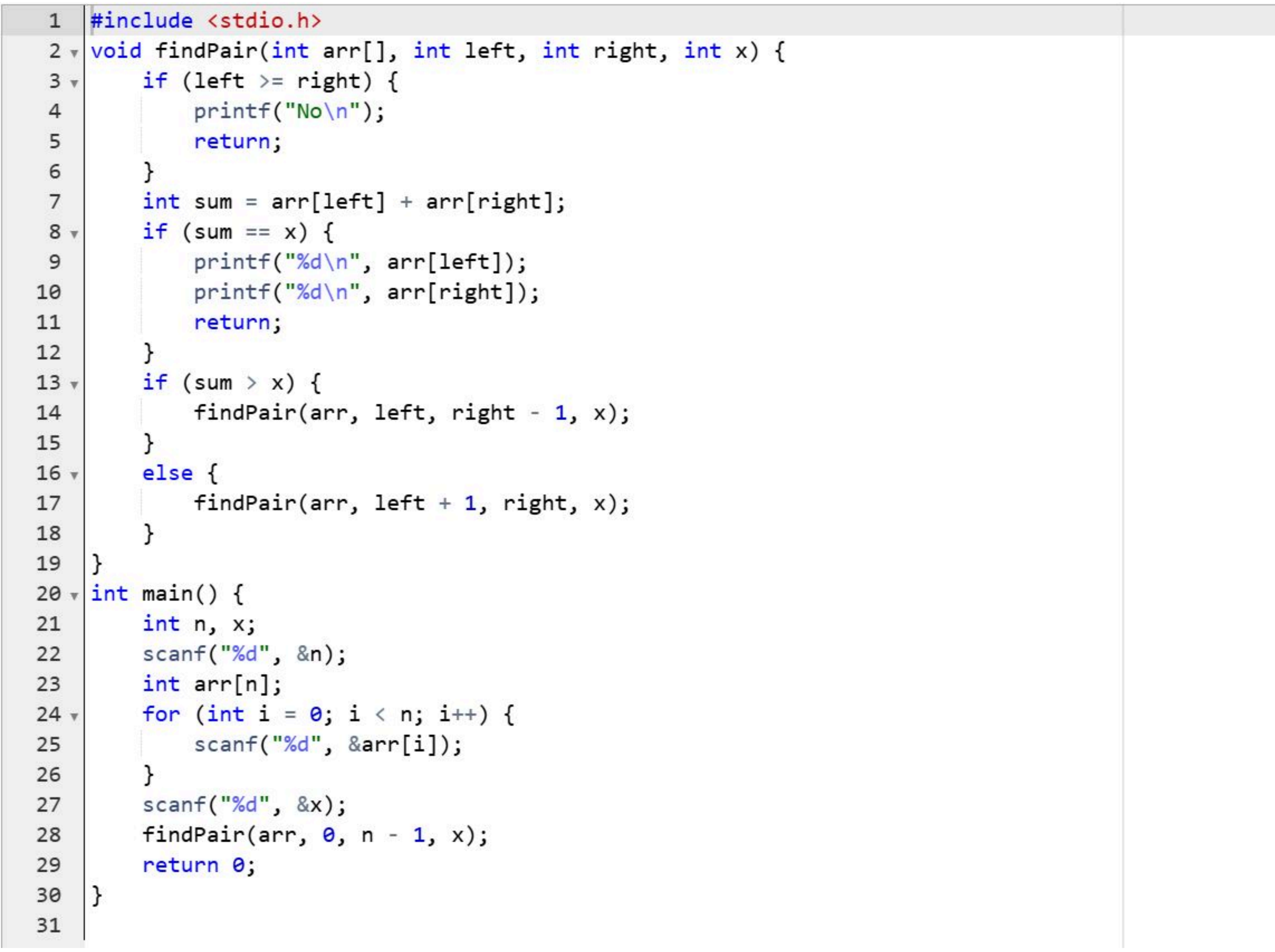
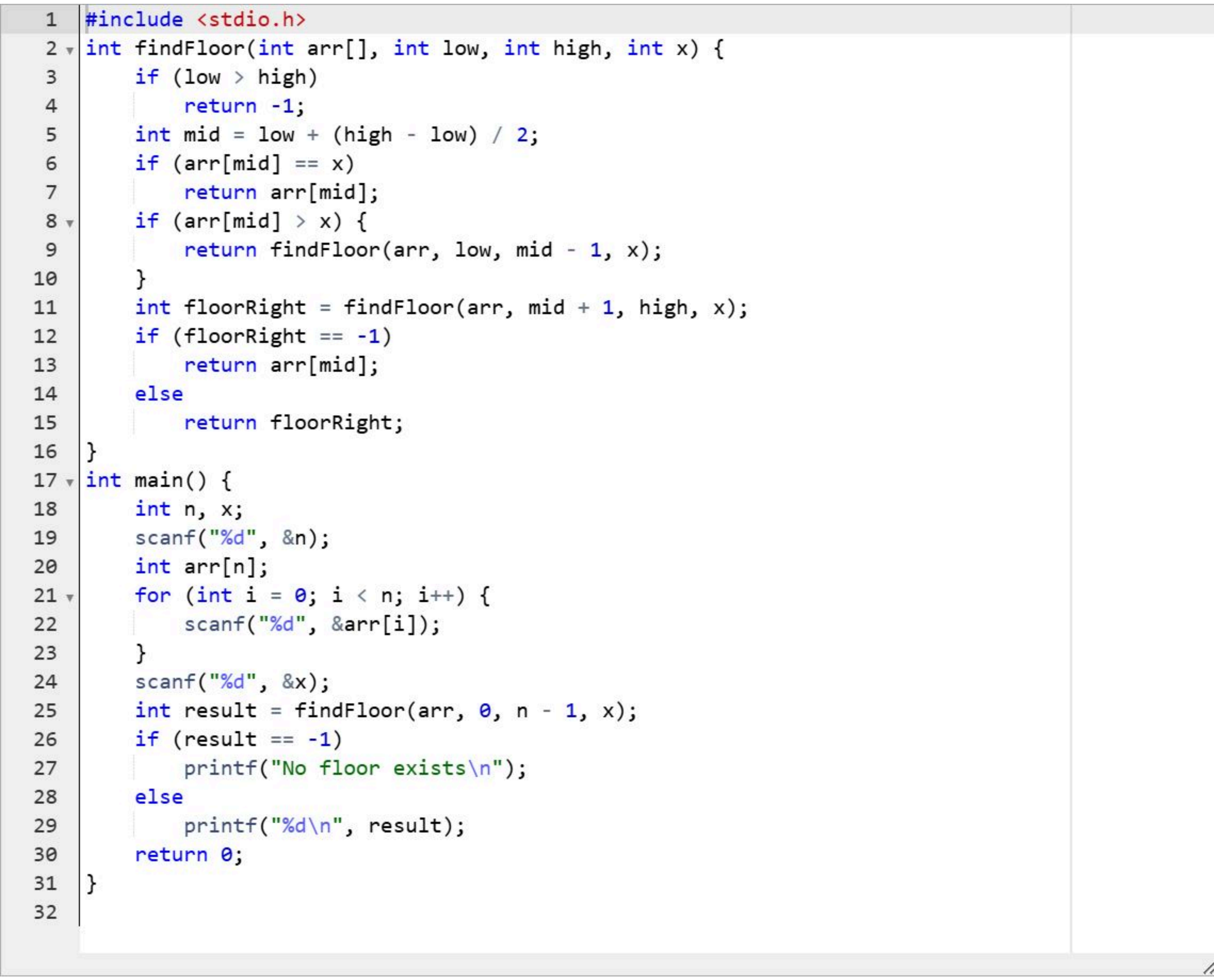


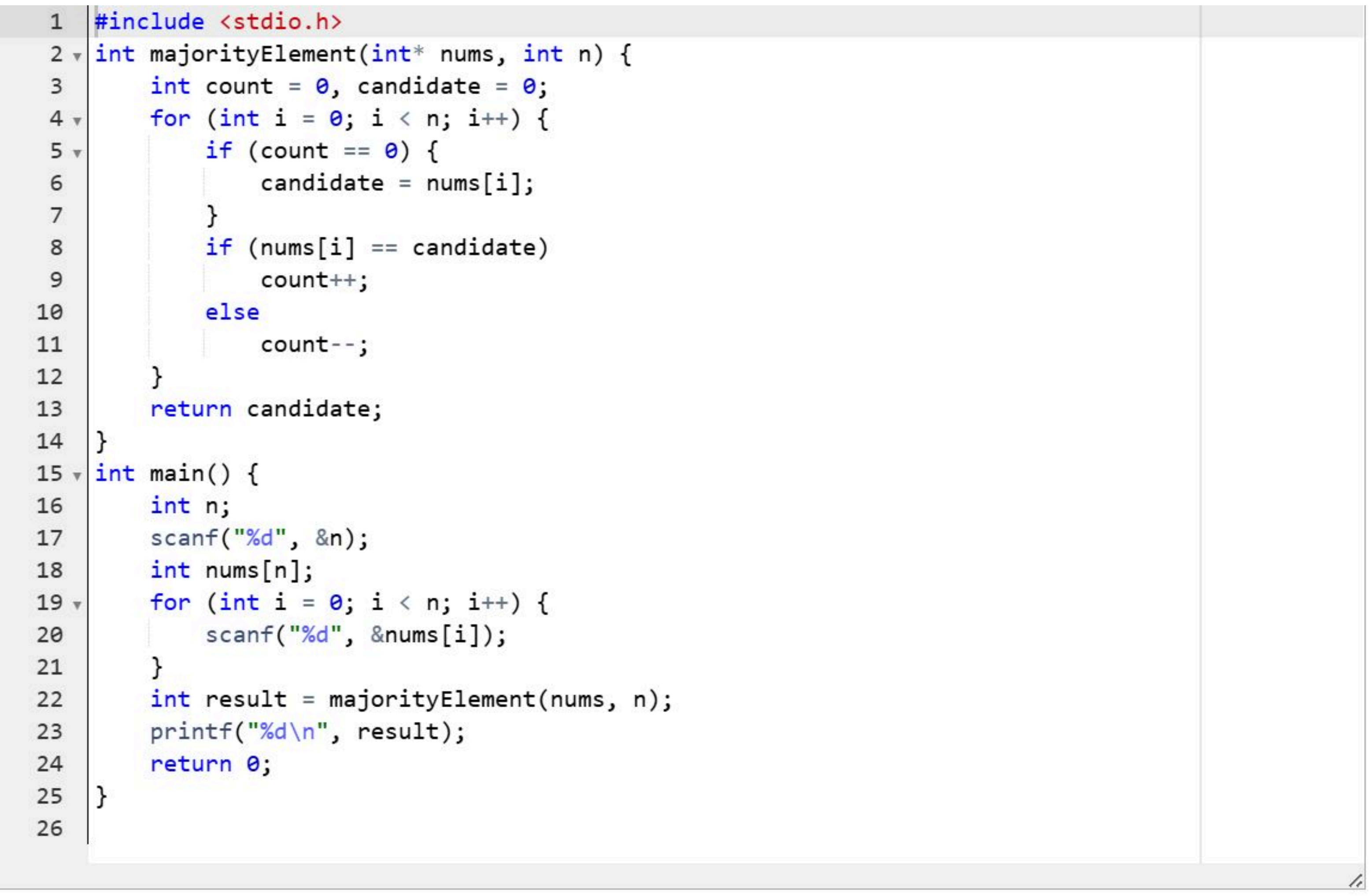
```c
#include <stdio.h>
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
int partition(int arr[], int low, int high) {
    int pivot = arr[high]; // choosing last element as pivot
    int i = (low - 1);
    for (int j = low; j < high; j++) {
        if (arr[j] <= pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
int main() {
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    quickSort(arr, 0, n - 1);
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
```



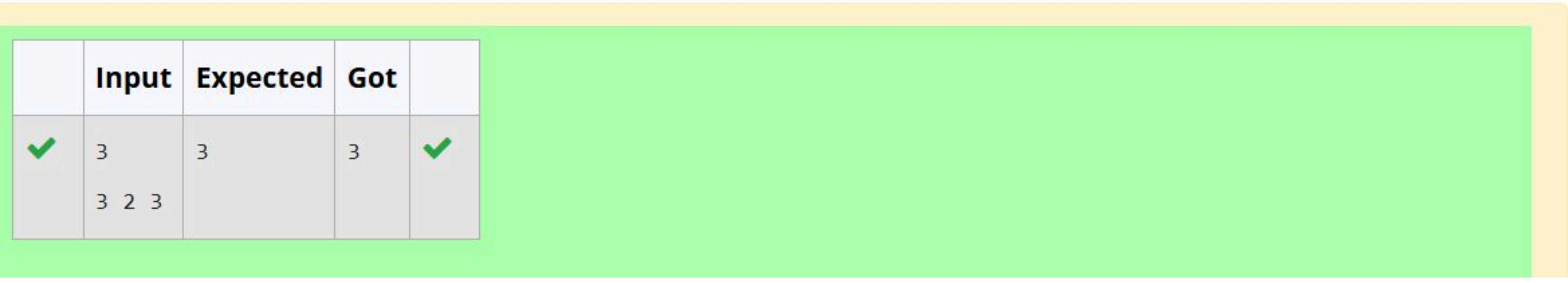
```c
#include <stdio.h>
void findPair(int arr[], int left, int right, int x) {
    if (left >= right) {
        printf("No\n");
        return;
    }
    int sum = arr[left] + arr[right];
    if (sum == x) {
        printf("%d\n", arr[left]);
        printf("%d\n", arr[right]);
        return;
    }
    if (sum > x) {
        findPair(arr, left, right - 1, x);
    }
    else {
        findPair(arr, left + 1, right, x);
    }
}
int main() {
    int n, x;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    scanf("%d", &x);
    findPair(arr, 0, n - 1, x);
    return 0;
}

```



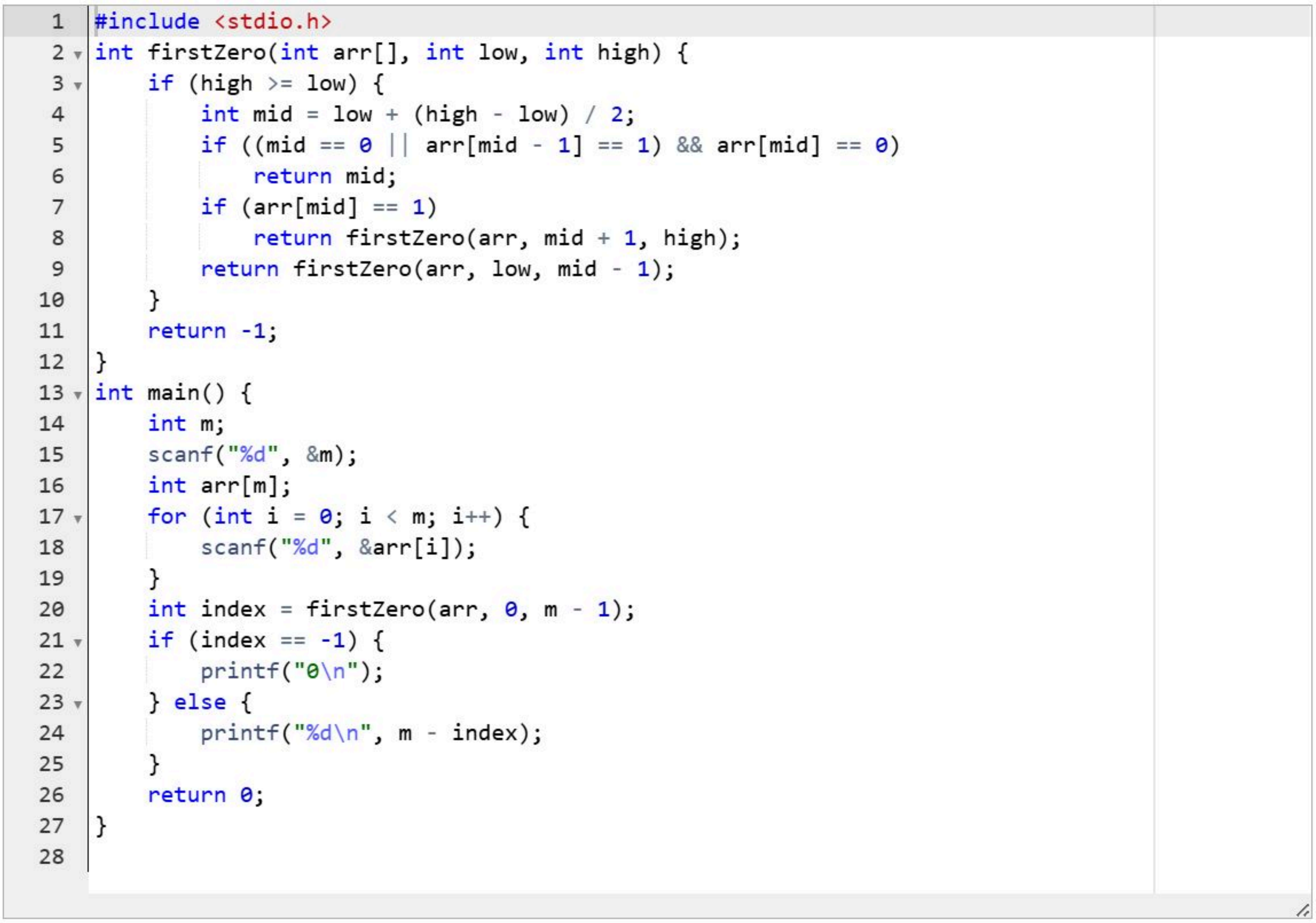
```c
#include <stdio.h>
int findFloor(int arr[], int low, int high, int x) {
    if (low > high)
        return -1;
    int mid = low + (high - low) / 2;
    if (arr[mid] == x)
        return arr[mid];
    if (arr[mid] > x) {
        return findFloor(arr, low, mid - 1, x);
    }
    int floorRight = findFloor(arr, mid + 1, high, x);
    if (floorRight == -1)
        return arr[mid];
    else
        return floorRight;
}
int main() {
    int n, x;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    scanf("%d", &x);
    int result = findFloor(arr, 0, n - 1, x);
    if (result == -1)
        printf("No floor exists\n");
    else
        printf("%d\n", result);
    return 0;
}
```



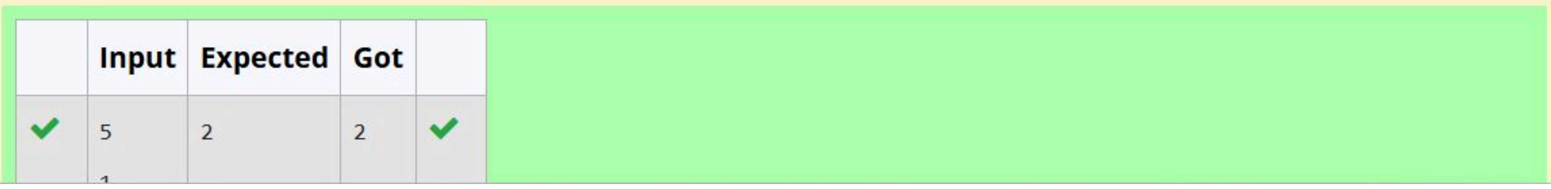
```c
#include <stdio.h>
int majorityElement(int* nums, int n) {
    int count = 0, candidate = 0;
    for (int i = 0; i < n; i++) {
        if (count == 0) {
            candidate = nums[i];
        }
        if (nums[i] == candidate)
            count++;
        else
            count--;
    }
    return candidate;
}
int main() {
    int n;
    scanf("%d", &n);
    int nums[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }
    int result = majorityElement(nums, n);
    printf("%d\n", result);
    return 0;
}

```



	Input	Expected	Got	
✓	3 3 2 3	3	3	✓



```c
#include <stdio.h>
int firstZero(int arr[], int low, int high) {
    if (high >= low) {
        int mid = low + (high - low) / 2;
        if ((mid == 0 || arr[mid - 1] == 1) && arr[mid] == 0)
            return mid;
        if (arr[mid] == 1)
            return firstZero(arr, mid + 1, high);
        return firstZero(arr, low, mid - 1);
    }
    return -1;
}
int main() {
    int m;
    scanf("%d", &m);
    int arr[m];
    for (int i = 0; i < m; i++) {
        scanf("%d", &arr[i]);
    }
    int index = firstZero(arr, 0, m - 1);
    if (index == -1) {
        printf("0\n");
    } else {
        printf("%d\n", m - index);
    }
    return 0;
}
```



	Input	Expected	Got	
✓	5 1	2	2	✓