```
class Solution {
    /*
    public List<Integer> findClosestElements(int[] arr, int k, int x) {
        Queue<Integer> q = new PriorityQueue<>((a, b) -> {
            int tmp = Math.abs(a - x) - Math.abs(b - x);
            return (tmp != 0) ? tmp : a - b;
            });

        for (int i : arr) {
            q.offer(i);
        }

        List<Integer> res = new ArrayList<>();
        while (!q.isEmpty() && res.size() != k) {
            res.add(q.poll());
        }

        Collections.sort(res);
        return res;
    }
    */

    public List<Integer> findClosestElements(int[] arr, int k, int x) {
        List<Integer> answer = new ArrayList<>();
        int left = 0, right = arr.length-1, mid = 0;
        while(left <= right){
            mid = left + (right-left)/2;
            if(arr[mid] == x) break;
            else if(arr[mid] < x) left = mid+1;
            else right = mid-1;
        }
        left = mid;
        right = mid+1;
        while(right - left <= k){
            if(left < 0) {
                //answer.add(arr[right++]);
                right++;
            } else if(right > arr.length-1){
                                        // Don't insert to front of list, it is very slow
                //answer.add(0, arr[left--]);
                left--;
            } else{
                if(Math.abs(x-arr[left]) <= Math.abs(x-arr[right])){
                    //answer.add(0, arr[left--]);
                    left--;
                } else{
                    //answer.add(arr[right++]);
                    right++;
                }
            }
        }
        while(++left < right) answer.add(arr[left]);
        return answer;
    }
```

```
}
```