Solution 1

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

#include <vector>
using namespace std;

int quickselectHelper(vector<int> array, int startIdx, int endIdx,
                      int position);

// Best: O(n) time | O(1) space
// Average: O(n) time | O(1) space
// Worst: O(n^2) time | O(1) space
int quickselect(vector<int> array, int k) {
  int position = k - 1;
  return quickselectHelper(array, 0, array.size() - 1, position);
}

int quickselectHelper(vector<int> array, int startIdx, int endIdx,
                      int position) {
  while (true) {
    if (startIdx > endIdx) {
      perror("Your Algorithm should never arrive here!");
      exit(1);
    }
    int pivotIdx = startIdx;
    int leftIdx = startIdx + 1;
    int rightIdx = endIdx;
    while (leftIdx <= rightIdx) {
      if (array[leftIdx] > array[pivotIdx] &&
          array[rightIdx] < array[pivotIdx]) {
        swap(array[leftIdx], array[rightIdx]);
      }
      if (array[leftIdx] <= array[pivotIdx]) {
        leftIdx++;
      }
      if (array[rightIdx] >= array[pivotIdx]) {
        rightIdx--;
      }
    }
    swap(array[pivotIdx], array[rightIdx]);
    if (rightIdx == position) {
      return array[rightIdx];
    } else if (rightIdx < position) {
      startIdx = rightIdx + 1;
    } else {
      endIdx = rightIdx - 1;
    }
  }
}

```