```
1 #include <iostream>
 2 #include <random>
 3 #include <vector>
 4 #include <chrono>
6 using namespace std;
 7
8 bool get line(const string& prompt, string& userinput){
       cout << prompt;</pre>
10
       getline(cin, userinput);
11
       return !userinput.empty();
12 }
13
14 int partition(vector<int> &arr, int lo, int hi, int
   pivot idx){
15
       int pivot value = arr[pivot idx];
       int left = lo - 1;
16
17
       int right = hi;
       swap(arr[pivot_idx], arr[right]);
18
19
       while(left < right){</pre>
20
           while(arr[++left] < pivot value) if(left == right)</pre>
    break;
21
           while(arr[--right] > pivot value) if(left == right
   ) break;
22
           if(left >= right) break;
23
           swap(arr[left], arr[right]);
24
25
       swap(arr[left],arr[hi]);
26
       return left;
27 }
28
29 void display array(const vector<int>& A){
30
       for(int e : A) cout << e << " ";</pre>
31
       cout << endl;
32 }
33
34 int quick_select_idx(vector<int> &arr, int k){
       unsigned int seed = chrono::steady clock::now().
35
   time since epoch().count();
36
       mt19937 gen(seed);
       int lo = 0, hi = arr.size() - 1;
37
38
       while(lo < hi){</pre>
           int pivot idx = uniform int distribution<int>{lo,
39
   hi}(gen);
40
           int new pivot idx = partition(arr, lo, hi,
   pivot idx);
41
           if(new pivot idx == k - 1) return new pivot idx;
           else if(new_pivot_idx < k - 1) lo = new_pivot_idx</pre>
42
   + 1;
43
           else hi = new pivot idx - 1;
```

```
44
45
       return lo;
46 }
47
48 vector<int> max k numbers(vector<int>& arr, int k){
       int starting idx = quick_select_idx(arr, arr.size() -
   k + 1);
50
       vector<int> result;
       for(int i = starting idx;i < arr.size();i++)</pre>
51
52
           result.push back(arr[i]);
53
       return result;
54 }
55
56 int main(){
       unsigned int seed = chrono::steady clock::now().
57
   time since epoch().count();
58
       mt19937 gen(seed);
59
       string userinput;
       while(get line("(part 1) Enter positive integer n: ",
60
   userinput)){
61
           int n = stoi(userinput);
62
           uniform int distribution<int>
   uniform int distribution(-100, 100);
63
64
           vector<int> A;
           for(int i = 0; i < n; i++)
65
               A.push back(uniform int distribution(gen));
66
67
68
           display array(A);
69
70
           string second input;
71
           get_line("Enter a number between 1 to n: ",
   second input);
72
           int kth = stoi(second input);
73
           int idx = quick select idx(A, kth);
74
           cout << A[idx] << endl;</pre>
75
       }
76
77
       while(get line("(part 2) Enter positive integer n: ",
   userinput)){
78
           int n = stoi(userinput);
79
           uniform int distribution<int>
   uniform int distribution(-100, 100);
80
81
           vector<int> A;
82
           for(int i = 0; i < n; i++)
83
               A.push back(uniform int distribution(gen));
84
85
           display_array(A);
86
```

```
File - /home/sergio/Desktop/lab3_cecs328/main.cpp
              string second_input;
 87
 88
              get_line("Enter a number between 1 to n: ",
     second_input);
              int kth = stoi(second_input);
 89
              auto result = max_k_numbers(A, kth);
 90
 91
              display_array(result);
         }
 92
 93 }
 94
```