```
#include <iostream>
#include <vector>

template <typename T>
int find(const std::vector<T> &data, const T key, int k, int lb, int ub) {
  if (ub - lb < k) {
    for (int i = lb; i < ub && data[i] <= key; ++i)
      if (data[i] == key)
        return i;
    return data.size();
  }
  int periods[k - 1];
  for (int i = 1; i < k; ++i)
    periods[i - 1] = lb * i + ub * (k - i) / k;
  if (key < data[periods[0]])
    return find(data, key, k, lb, periods[0]);
  if (key == data[periods[0]])
    return periods[0];
  for (int i = 1; i < k - 1; ++i) {
    if (key < data[periods[i]])
      return find(data, key, k, periods[i - 1] + 1, periods[i]);
    if (key == data[periods[i]])
      return periods[i];
  }
  return find(data, key, k, periods[k - 2] + 1, ub);
}

template <typename T> int find(const std::vector<T> &data, const T key, int k) {
  return find(data, key, k, 0, data.size());
}

int main() {
  std::vector<int> data = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19};
  int key;
  int k;

  std::cout << "input key value: ";
  std::cin >> key;

  std::cout << "input k: ";
  std::cin >> k;

  int result = find(data, key, k);

  if (result == data.size())
    std::cout << "cannot find " << key << "\n";
  else
    std::cout << "find " << key << " in the index of " << result << "\n";
  return 0;
}
```