find_max.cpp

```cpp
// Analyze unordered find_max
// 1. fill a vector with n randomly chosen values
// 2. see how many operations are required to find the maximum value
// Jon Beck
// 17 January 2022

#include <ctime>
#include <iostream>
#include <random>
#include <vector>
using namespace std;

/**
 * use linear search to find the largest value in the array
 * @param array the array of values to search
 * @return the largest value in the array
 */
unsigned find_max(const vector <unsigned>& array);

int main(int argc, char* argv[])
{
  if (argc != 2)
  {
    cerr << "Usage: " << argv[0] <<
       " n where n is the number of values in the array" << endl;
    return 1;
  }

  srand(static_cast<unsigned>(time(nullptr)));
  unsigned number_of_values = static_cast<unsigned>(stoul(argv[1]));

  vector<unsigned> values;
  for (unsigned count = 0; count < number_of_values; count++)
  {
    values.push_back(static_cast<unsigned>(rand()));
  }

  cout << "Maximum value: " << find_max(values) << endl;
  return 0;
}

unsigned find_max(const vector<unsigned>& array)
{
  unsigned max_index = array.at(0);
  size_t n = array.size();

  for (size_t index = 1; index < n; index++)
  {
    max_index = array.at(index) > max_index ? array.at(index) : max_index;
  }
  return max_index;
}
```