search.cpp

```cpp
#include "search.h"
#include <string>
#include <iostream>

void rand_mass(int* mass, unsigned n, int min, int max)
{
        for (int i = 0; i < n; i++)
                mass[i] = min + rand() % max;
}

string search(int* mass, int n, int key)
{
        string result = "Результат поиска по ключу (" + to_string(key) + ") : ";
        for (int i = 0; i < n; i++) {
                if (mass[i] == key) {
                        result += to_string(i) + "  ";
                }
        }
        return result;
}

void input(int* mass, int n)
{
        for (int i = 0; i < n; i++) {
                cout << "Элемент с номером " << i << ": ";
                cin >> mass[i];
        }
}

void output(int* mass, int n)
{
        int k = 0;
        for (int i = 0; i < n; i++) {
                cout << mass[i] << " \t";
                k++;
                if (k == 10) {
                        cout << endl;
                        k = 0;
                }
        }
        cout << endl;
}
```

search.h

```cpp
#pragma once
#include <string>
#include <iostream>

using namespace std;

void rand_mass(int* mass, unsigned n, int min, int max);
string search(int* mass, int n, int key);
void input(int* mass, int n);
void output(int* mass, int n);
```

main.cpp

```cpp
#include <iostream>
#include "search.h"
#include <iostream>
#include <ctime>

using namespace std;

const int n = 10000;

int main()
{
        setlocale(LC_ALL, "Rus");
        int* mass = new int[n];
        int min = 0;
        int max = 1000;
        rand_mass(mass, n, min, max);
        cout << "Введенный массив : " << endl;
        output(mass, n);
        unsigned int start_time = clock(); // начальное время
        cout << search(mass, n, min + rand() % max) << endl;
        unsigned int end_time = clock(); // конечное время
        cout << "Время поиска по ключу : " << ((float)end_time - start_time) / CLOCKS_PER_SEC << " секунды " << endl;
        input(mass, 10);
        cout << "Введенный массив : ";
        output(mass, 10);
        cout << search(mass, 10, 5) << endl;
        delete(mass);
}
```