**Отчет**

**Алгоритм: std::remove\_if**

**Задача: удалить из случайного массива числа, меньшие 18**

**Код программы:**

#include <iostream>

#include <vector>

#include <algorithm>

#include <execution>

#include <ctime>

#include <random>

using namespace std;

vector<int> random\_numbers(long size) {

vector<int> numbers;

for (int i = 0; i < size; i++) {

numbers.push\_back(rand() % 101);

}

return numbers;

}

bool remove\_number(int number) {

return number < 18;

}

double sequential\_execution(long size) {

auto mas = random\_numbers(size);

auto start\_time = clock();

std::remove\_if(mas.begin(), mas.end(), remove\_number);

auto end\_time = clock();

double sequent\_time = (double)(end\_time - start\_time) / CLOCKS\_PER\_SEC;

std::cout << "Sequent time :" << sequent\_time << endl;

return sequent\_time;

}

double parallel\_execution(long size) {

auto mas = random\_numbers(size);

auto start\_time = clock();

std::remove\_if(std::execution::par, mas.begin(), mas.end(), remove\_number);

auto end\_time = clock();

double parallel\_time = (double)(end\_time - start\_time) / CLOCKS\_PER\_SEC;

std::cout << "Parallel time :" << parallel\_time << endl;

return parallel\_time;

}

int main() {

std::cout << "\nTask size : " << 1000000 << endl;

double sequent\_time = sequential\_execution(1000000);

double parallel\_time = parallel\_execution(1000000);

double acceleration\_factor = sequent\_time / parallel\_time;

std::cout << "Acceleration : " << acceleration\_factor << endl;

std::cout << "\nTask size : " << 100000000 << endl;

sequent\_time = sequential\_execution(100000000);

parallel\_time = parallel\_execution(100000000);

double acceleration\_factor = sequent\_time / parallel\_time;

std::cout << "Acceleration : " << acceleration\_factor << endl;

}

**Результат:**

