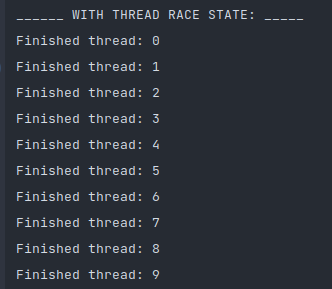
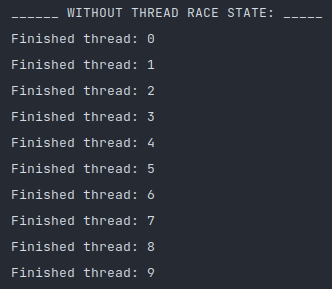
Задание 1

#include <iostream>  
#include <thread>  
#include <mutex>  
#include <sstream>  
#include <vector>  
  
  
*static void* to\_cout(*int* id) {  
 std::cout << "Finished thread: " << id << '\n';  
}  
  
*struct* pcout : *public* std::*stringstream* {  
 *static inline* std::mutex cout\_mutex;  
 ~pcout() {  
 std::lock\_guard<std::mutex> l{cout\_mutex};  
 std::cout << rdbuf();  
 std::cout.flush();  
 }  
};  
  
*static void* to\_pcout(*int* id) {  
 pcout{} << "Finished thread: " << id << '\n';  
}  
  
*void* race(std::vector<std::thread>& vector) {  
 std::cout << "\_\_\_\_\_\_ WITH THREAD RACE STATE: \_\_\_\_\_\n";  
 vector.clear();  
 *for* (*size\_t* i{0}; i < 10; ++i) {  
 vector.emplace\_back(to\_cout, i);  
 }  
 *for* (*auto* &t: vector) { t.join(); }  
}  
  
*void* no\_race(std::vector<std::thread>& vector) {  
 std::cout << "\_\_\_\_\_\_ WITHOUT THREAD RACE STATE: \_\_\_\_\_\n";  
 vector.clear();  
 *for* (*size\_t* i{0}; i < 10; ++i) {  
 vector.emplace\_back(to\_pcout, i);  
 }  
 *for* (*auto* &t: vector) { t.join(); }  
}  
  
*int* main() {  
 std::vector<std::thread> vector;  
 race(vector);  
 no\_race(vector);  
}





Задание 2

#include <iostream>  
#include <thread>  
#include <mutex>  
#include <sstream>  
#include <vector>  
#include <fstream>  
  
std::*ofstream* out\_("test.txt");  
  
*void* to\_cout(*int* id) {  
 out\_ << "Thread: " << id << std::endl;  
}  
  
*struct* pcout : *public* std::*stringstream* {  
 std::mutex mutex;  
 ~pcout() {  
 std::lock\_guard<std::mutex> l{mutex};  
 out\_ << rdbuf();  
 out\_.flush();  
 }  
};  
  
*void* to\_pcout(*int* id) {  
 pcout{} << "Thread: " << id << std::endl;  
}  
  
*void* race() {  
 out\_ << "\_\_\_\_\_\_ WITH THREAD RACE STATE: \_\_\_\_\_\n";  
 std::vector<std::thread> vector(0);  
 *for* (*size\_t* i{0}; i < 10; ++i) {  
 vector.emplace\_back(to\_cout, i);  
 }  
 *for* (*auto* &t: vector) { t.join(); }  
}  
  
*void* no\_race() {  
 out\_ << "\_\_\_\_\_\_ WITHOUT THREAD RACE STATE: \_\_\_\_\_\n";  
 std::vector<std::thread> vector(0);  
 *for* (*size\_t* i{0}; i < 10; ++i) {  
 vector.emplace\_back(to\_pcout, i);  
 }  
 *for* (*auto* &t: vector) { t.join(); }  
}  
  
*int* main() {  
 race();  
 no\_race();  
 *return* 0;  
}

