ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ “ВЫСШАЯ ШКОЛА ЭКОНОМИКИ”»

**ДОМАШНЕЕ ЗАДАНИЕ №3**

Работа студента 2 курса бакалавариата группы БПИ-194

по предмету «Архитектура вычислительных систем»

**Выполнил:**

Сайфудинов Нурсултан Аскатович,

БПИ194-1

**Преподаватель:**

Доктор технических наук,

Профессор

Легалов А. И

**Москва 2020**

**Задание:**

Задача о Пути Кулака. На седых склонах Гималаев стоят два древних буддистских монастыря: Гуань-Инь и Гуань-Янь. Каждый год в день сошествия на землю боддисатвы Араватти монахи обоих монастырей собираются на совместное празднество и показывают свое совершенствование на Пути Кулака. Всех соревнующихся монахов разбивают на пары, победители пар бьются затем между собой и так далее, до финального поединка. Монастырь, монах которого победил в финальном бою, забирает себе на хранение статую боддисатвы. Реализовать многопоточное приложение, определяющего победителя. В качестве входных данных используется массив, в котором хранится количество энергии Ци каждого монаха. При решении использовать принцип дихотомии.

**Решение:**

При запуске программы программа просит ввести пользователя количество монахов, участвующих в турнире. После получения информации о количестве участников, программа создает массив с энергией Ци каждого монаха. Далее программа создает два массива с монахами каждого из монастырей и разделяет энергию Ци среди них.

В дальнейшем программа имитирует бой сравнивая количество энергии Ци каждого монаха с соперником. Так реализовывается турнирная сетка, где победивший идет дальше и сражается с другим победителем.

Вывод данных осуществляется в консоль.

Так монахи сражаются друг с другом, до тех пор, пока не останется один победитель с наибольшим количеством энергии Ци. Монастырь, монах которого оказался сильнейшим, забирает себе статуэтку.

При вводе количества участников, обязательно должно быть введено число, и оно должно быть больше нуля. Для этого была разработана проверка введенного значения. Для генерации массивов использовался псевдорандом.

**Описание методов:**

* **vector<int> GenerateArray (int size) –** генерация массива длинойsize
* **void Fight(vector<int> &array1, vector<int> &array2) –** имитация турнира с боями между монастрыями.

Текст программы приведен ниже:

#include <iostream>

#include <string>

#include <fstream>

#include <sstream>

#include <map>

#include <unordered\_map>

#include <algorithm>

#include <thread>

#include <cstdlib>

#include <ctime>

#include <omp.h>

using namespace std;

vector<int> GenerateArray(int size)

{

srand(time(0));

vector<int> array;

for (size\_t i = 0; i < size; i++)

{

array.push\_back(1 + rand() % 100);

}

return array;

}

int getIndex(vector<int> v, int K)

{

auto it = find(v.begin(), v.end(), K);

if (it != v.end())

{

int index = it - v.begin();

return index;

}

else {

return -1;

}

}

void FightMode(int guanin, int guanyan, vector<int>& array1, vector<int>& array2)

{

srand(time(0));

#pragma omp critical

{

if (guanin > guanyan)

{

array2[getIndex(array2, guanyan)] = 0;

}

else if (guanin < guanyan)

{

array1[getIndex(array1, guanin)] = 0;

}

else if (guanin == guanyan)

{

if ((rand() % 2) == 0)

{

array2[getIndex(array2, guanyan)] = 0;

}

else

{

array1[getIndex(array1, guanin)] = 0;

}

}

}

}

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Введите пожалуйста количество участников в турнире!(положительное больше нуля!)" << endl;

vector<int> guanin;

vector<int> guanyan;

vector<int> energy;

string s;

cin >> s;

char\* end;

long i = strtol(s.c\_str(), &end, 10);

if (\*end == '\0')

{

if (i > 0)

{

energy = GenerateArray(i);

for (size\_t i = 0; i < energy.size(); i++)

{

if ((i % 2) == 0)

{

guanin.push\_back(energy[i]);

}

else

{

guanyan.push\_back(energy[i]);

}

}

while (guanin.size() != 0 && guanyan.size() != 0)

{

for (size\_t i = 0; i < guanin.size(); i++)

{

if (guanin[i] == 0)

{

guanin.erase(guanin.begin() + i);

--i;

}

}

for (size\_t i = 0; i < guanyan.size(); i++)

{

if (guanyan[i] == 0)

{

guanyan.erase(guanyan.begin() + i);

--i;

}

}

int size = guanyan.size() < guanin.size() ? guanyan.size() : guanin.size();

omp\_set\_num\_threads(size > 100 ? 100 : (size + 8) / 8);

#pragma omp parallel

{

#pragma omp for

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

{

FightMode(guanin[i], guanyan[i], guanin, guanyan);

}

}

}

if (guanin.size() != 0)

{

cout << "Победил монастырь Гуань-Инь. Их Путь Кулака более совершенен. Они забирают себе на хранение статую боддисатвы Араватти." << endl;

}

else

{

cout << "Победил монастырь Гуань-Янь. Их Путь Кулака более совершенен. Они забирают себе на хранение статую боддисатвы Араватти." << endl;

}

}

else

{

cout << "К сожалению вы вввели некорректные данные!" << endl;

}

}

else

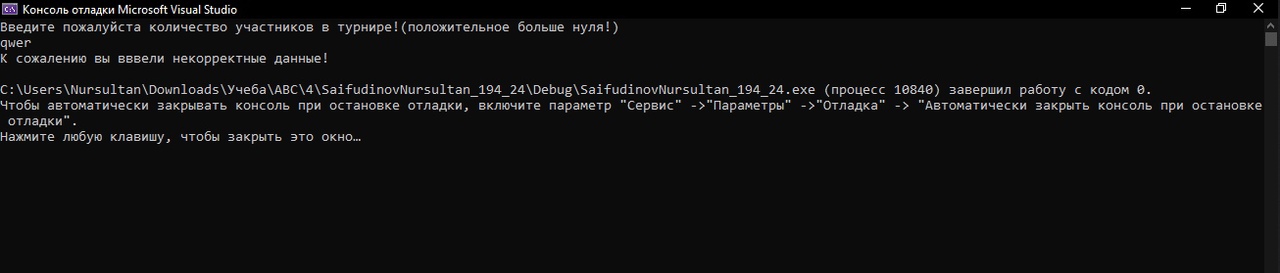
{

cout << "К сожалению вы вввели некорректные данные!" << endl;

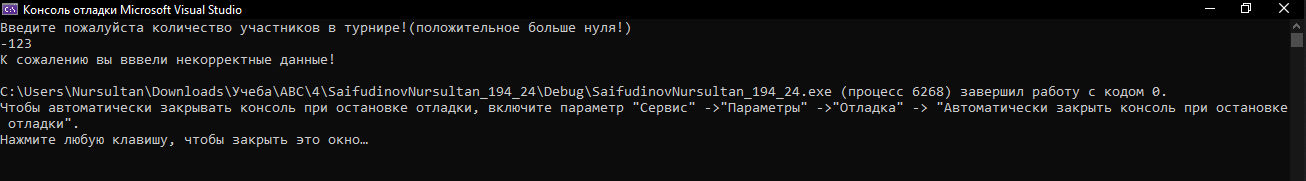
}

}

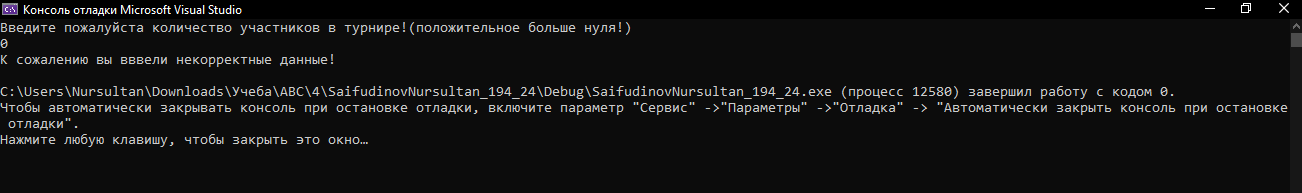
**Результаты разработки:**



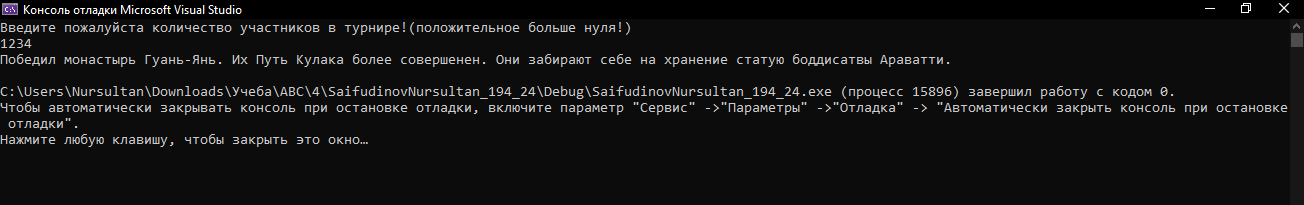
**TEST1**

****

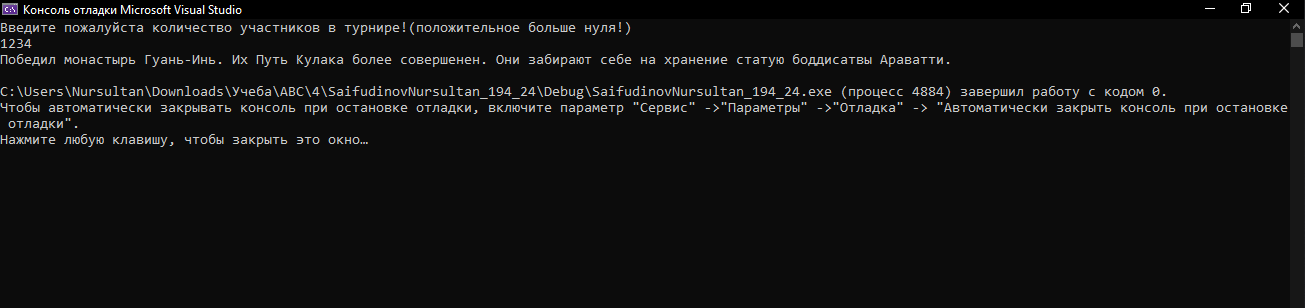
**TEST2**

****

**TEST3**

****

**TEST4**

****

**TEST5**