

Пояснительная записка к домашнему заданию №3  
**Многопоточное приложение,  
реализующее задачу Пути Кулака**

НИУ ВШЭ, ДПИ  
Тарасюк Инна Валерьевна, БПИ192(1)  
Вариант 24

## 1 Условие задания

**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.

## 2 Описание работы программы

### 2.1 Выбор модели построения многопоточного приложения

Способ организации потоков – древовидная структура или сети слияния, на этом основан, в частности, метод дихотомии.

## 3 Работа программы

### 3.1 Переменные

1. `n` - количество жителей двух монастырей; значение задаётся пользователем;
2. `max_energy` - максимально допустимое значение энергии Ци; задаётся пользователем;
3. `team` - массив размера `n`, состоящий из жителей двух монастырей.
4. `team_first` - массив, полученный путем деления исходного массива на две части; содержит первую половину исходного массива;
5. `team_second` - массив, полученный путем делением исходного массива на две части; содержит вторую половину исходного массива;
6. `energy1` - энергия Ци представителя первой команды;
7. `energy2` - энергия Ци представителя второй команды;
8. `winners` - массив победителей(-я);
9. `result` - энергия Ци победившего монаха.

### 3.2 Функции

Название функции	Тип возвращаемого значения	Входные параметры
main	int	-
input	int	-
threadFunction	vector<int> team, int energy	void

1. main() - функция, в которой происходит: 1) считывание введенного пользователем значения, обозначающего количество монахов; 2) заполнение массива монахов случайными числами в диапазоне [1;max\_energy] - энергией Ци; 3) вызов функции threadFunction для реализации поставленной задачи; 4) вывод информации о жителях монастырей и победившем(-их) монахе;
2. int input() - функция, в которой происходит считывание введенной пользователем строки и осуществляется проверка её на корректность (корректным считается ввод числа);
3. void threadFunction(std::vector<int> team, int winner) - функция, в которой происходит: 1) деление (переданного первым параметром) массива team на два других массива - team\_first и team\_second; 2) запуск двух новых потоков thr1 и thr2 для одного исходного, то есть двукратный вызов threadFunction - отдельно для team\_first и team\_second до тех пор, пока размер передаваемого массива не станет <= 2; 3) определение победителя пары путём сравнения значений энергии Ци (energy1 сравнивается с energy2); 4) ожидание завершения thr1 и thr2 с помощью join() и свертка рекурсии; так продолжается до последнего потока, в котором определяется итоговый победитель.

### 3.3 Алгоритм

Программе на вход подается число n и max\_energy, осуществляется их проверка на корректность с помощью функции input(). Создаётся массив размера n, который заполняется случайными числами в диапазоне [1; max\_energy] - значения энергии Ци. Вызывается функция threadFunction, определяющая значение энергии победителя методом дихотомии(см. 3.2.3). Сведения о всех сражавшихся монахах и победителе(-ях) среди них выводятся на экран.

## 4 Входные данные

1. n - размер массива для энергий монахов;
2. max\_energy - максимально возможная энергия монаха.

## 4.1 Ограничения

Представленные ниже ограничения были введены для удобства и ввиду здравого смысла.

Переменная	Минимальное значение	Максимальное значение
n (количество монахов)	1	100
max_energy	1	-

## 5 Выходные данные

Два вида вывода:

1. "Monk  $\langle i \rangle$  won with qi energy  $\langle value \rangle$   
Guan-yin(yang) got bodhisattva statue".  
Указание монастыря, который забирает статую.
2. "The battle ended in a draw  
Monk  $\langle i1 \rangle$  won with qi energy  $\langle value \rangle$   
Monk  $\langle i2 \rangle$  won with qi energy  $\langle value \rangle$   
..."  
Монастырь, забирающий статую, не определён, так как у  $\geq 2$  монахов была одинаковая наибольшая энергия Ци.

## 6 Тестирование программы

### 6.1 Некорректные данные

```
C:\Users\kysar\CLionProjects\untitled\cmake-build-debug
Input a number of monks <= 100:
xcvbnm
Incorrect input. Try again.
Input a number of monks <= 100:
1000000000000
Incorrect input. Try again.
Input a number of monks <= 100:
-100
Input a number of monks <= 100:
11
Input max energy value:
8
Guan-yin. Monk 1.Qi energy 1
Guan-yin. Monk 2.Qi energy 5
Guan-yin. Monk 3.Qi energy 7
Guan-yin. Monk 4.Qi energy 1
Guan-yin. Monk 5.Qi energy 6
Guan-yin. Monk 6.Qi energy 3
Guan-yang. Monk 7.Qi energy 5
Guan-yang. Monk 8.Qi energy 1
Guan-yang. Monk 9.Qi energy 4
Guan-yang. Monk 10.Qi energy 4
Guan-yang. Monk 11.Qi energy 2

Monk 3 won with qi energy 7
Guan-yin got bodhisattva statue.
Process finished with exit code 0
```

## 6.2 Корректные данные

```
my_project x
C:\Users\kysar\CLionProjects\untitled\cmake-bu
Input a number of monks <= 100:
15
Input max energy value:
20
Guan-yin. Monk 1.Qi energy 2
Guan-yin. Monk 2.Qi energy 4
Guan-yin. Monk 3.Qi energy 9
Guan-yin. Monk 4.Qi energy 2
Guan-yin. Monk 5.Qi energy 18
Guan-yin. Monk 6.Qi energy 8
Guan-yin. Monk 7.Qi energy 8
Guan-yin. Monk 8.Qi energy 8
Guan-yang. Monk 9.Qi energy 4
Guan-yang. Monk 10.Qi energy 8
Guan-yang. Monk 11.Qi energy 12
Guan-yang. Monk 12.Qi energy 11
Guan-yang. Monk 13.Qi energy 14
Guan-yang. Monk 14.Qi energy 2
Guan-yang. Monk 15.Qi energy 9

Monk 5 won with qi energy 18
Guan-yin got bodhisattva statue.
Process finished with exit code 0
```



```
C:\Users\kysar\CLionProjects\untitled\cmak
Input a number of monks <= 100:
10
Input max energy value:
2
Guan-yin. Monk 1.Qi energy 2
Guan-yin. Monk 2.Qi energy 1
Guan-yin. Monk 3.Qi energy 2
Guan-yin. Monk 4.Qi energy 1
Guan-yin. Monk 5.Qi energy 1
Guan-yang. Monk 6.Qi energy 1
Guan-yang. Monk 7.Qi energy 1
Guan-yang. Monk 8.Qi energy 1
Guan-yang. Monk 9.Qi energy 2
Guan-yang. Monk 10.Qi energy 1

The battle ended in a draw
Monk 1 won with qi energy 2
Monk 3 won with qi energy 2
Monk 9 won with qi energy 2

Process finished with exit code 0
```

## 7 Текст программы

```
#include <iostream>
#include <thread>
```

```

#include <vector>
#include <ctime>
#include <random>

void threadFunction(std::vector<int> team, int &winner) {
    int energy1 = 0;
    int energy2 = 0;
    int sizeTeam = team.size();
    if (sizeTeam > 2) {
        size_t size = sizeTeam / 2 + sizeTeam % 2;
        std::vector<int> team_first(size);
        std::vector<int> team_second(size);
        for (size_t i = 0; i < size; ++i) {
            team_first[i] = team[i];
            team_second[i] = team[sizeTeam - 1 - i];
        }
        if (sizeTeam % 2 == 1) team_second[size - 1] = 0;
        std::thread thr1(threadFunction, team_first, std::ref(energy1));
        std::thread thr2(threadFunction, team_second, std::ref(energy2));
        thr1.join();
        thr2.join();
        (energy1 > energy2) ? winner = energy1 : winner = energy2;
    } else {
        (team[0] > team[1]) ? winner = team[0] : winner = team[1];
    }
}

int input() {
    int n;
    std::cin >> n;
    bool test = true;
    do {
        if (!(test = std::cin.good())) {
            std::cout << "Incorrect input. Try again." << std::endl;
            std::cin.clear();
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
        }
    } while (!test);
    return n;
}

int main() {
    int n;
    int max_energy;
    std::vector<int> winners;
    do {

```



```

        std::cout << "Input a number of monks <= 100:" << std::endl;
        n = input();
    } while (n <= 0 || n >= 100);
    do {
        std::cout << "Input max energy value: " << std::endl;
        max_energy = input();
    } while (max_energy <= 0);
    std::vector<int> team(n);
    std::mt19937 gen;
    gen.seed(time(0));
    for (size_t i = 0; i < n; ++i) {
        team[i] = gen() % max_energy + 1;
    }
    int result = 0;
    threadFunction(team, result);
    for (size_t i = 0; i < n; ++i) {
        if (i < (n / 2 + n % 2)) {
            std::cout << "Guan-yin. Monk " << i + 1
                << ".Qi energy " << team[i] << "\n";
        } else {
            std::cout << "Guan-yang. Monk " << i + 1
                << ".Qi energy " << team[i] << "\n";
        }
        if (team[i] == result)
            winners.push_back(i + 1);
    }
    if (winners.size() > 1) {
        std::cout << "\nThe battle ended in a draw" << "\n";
        for (size_t i = 0; i < winners.size(); ++i) {
            std::cout << "Monk " << winners[i] << " won with qi energy "
                << result << "\n";
        }
    } else {
        std::cout << "\n" << "Monk " << winners[0]
            << " won with qi energy " << result << "\n";
        (winners[0] > (n / 2 + n % 2)) ? std::cout
            << "Guan-yang got bodhisattva statue." : std::cout
            << "Guan-yin got bodhisattva statue.";
    }
    return 0;
}

```

## Список литературы

1. информация о потоках <https://habr.com/ru/post/279653/>
2. информация о потоках <https://docs.microsoft.com/ru-ru/cpp/standard-library/thread-class?view=msvc-160viewFallbackFrom=vs-2019>
3. информация о дихотомии <https://habr.com/ru/company/otus/blog/504310/>
4. информация о дихотомии <http://www.machinelearning.ru/wiki/index.php?title=>
5. <http://softcraft.ru/>