

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. Модель построения основана на параллельных секциях в OpenMP.

## 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;

### 3.2 Функции

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

1. main()
2. int input()
3. void threadFunction(std::vector<int> team, int l, int v)

### 3.3 Алгоритм

Программе на вход подается число  $n$  и  $\text{max\_energy}$ , осуществляется их проверка на корректность с помощью функции `input()`. Создаётся массив размера  $n$ , который заполняется случайными числами в диапазоне  $[1; \text{max\_energy}]$  - значения энергии Ци. Вызывается функция `threadFunction`, определяющая, у какого монаха энергия больше. Сведения о всех сражавшихся монахах и победителе(-ях) среди них выводятся на экран.

## 4 Входные данные

1.  $n$  - размер массива для энергий монахов;
2.  $\text{max\_energy}$  - 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
$\text{max\_energy}$	1	-

## 5 Тестирование программы

### 5.1 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
```

## 5.2 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
```

## 6 Текст программы

## 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/>