**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

СОГЛАСОВАНО

Профессор департамента

программной инженерии факультета компьютерных наук, к.т.н.

\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Авдошин

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

УТВЕРЖДАЮ

|  |
| --- |
| Академический руководитель образовательной программы «Программная инженерия», к.т.н. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

Подп. и дата

**КРИПТОСИСТЕМА РАБИНА**

**(RABIN CRYPTOSYSTEM)**

**БИБЛИОТЕКА RabinLib**

**Руководство программиста**

Инв. № дубл.

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729. 501430-01 33 01-1**

Взам. инв. №

Исполнитель:

студент группы БПИ 165

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. С. Чуев

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

Подп. и дата

Инв. № подл

RU.17701729.

501430-01

33 01-1

**2017**

УТВЕРЖДЕН  
RU.17701729. 501430-01 33 01-1-ЛУ

**КРИПТОСИСТЕМА РАБИНА**

**(RABIN CRYPTOSYSTEM)**

**БИБЛИОТЕКА RabinLib**

**Руководство программиста**

**RU.17701729. 501430-01 33 01-1**

Подп. и дата

Инв. № дубл.

Взам. инв. №

**Листов 18**

Подп. и дата

Инв. № подл

RU.17701729.

501430-01

33 01-1

**2017**

СОДЕРЖАНИЕ

[1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ БИБЛИОТЕКИ 4](#_Toc482987873)

[1.1. Назначение библиотеки 4](#_Toc482987874)

[1.2. Функции, выполняемые библиотекой 4](#_Toc482987875)

[1.3. Условия необходимые для выполнения библиотеки 5](#_Toc482987876)

[1.3.1. Требования к составу и параметрам технических средств 5](#_Toc482987877)

[1.3.2. Требования к программным средствам 5](#_Toc482987878)

[2. ХАРАКТЕРИСТИКИ ПРОГРАММЫ 6](#_Toc482987879)

[2.1. Временные характеристики 6](#_Toc482987880)

[2.2. Средства контроля правильности выполнения 6](#_Toc482987881)

[3. ОБРАЩЕНИЕ К БИБЛИОТЕКЕ 7](#_Toc482987882)

[3.1. Задание ключа 7](#_Toc482987883)

[3.1.1. Генерация ключей 7](#_Toc482987884)

[3.1.2. Задание существующих ключей 7](#_Toc482987885)

[3.2. Кодирование 7](#_Toc482987886)

[3.2.1. Кодирование числа 8](#_Toc482987887)

[3.2.2. Кодирование объекта наследника объекта Stream 8](#_Toc482987888)

[3.2.3. Кодирование файла 8](#_Toc482987889)

[3.3. Декодирование 8](#_Toc482987890)

[3.3.1. Декодирование числа 8](#_Toc482987891)

[3.3.2. Декодирование объекта наследника объекта Stream 8](#_Toc482987892)

[3.3.3. Декодирование файла 9](#_Toc482987893)

[3.4. Получение отчета о выполняемом процессе 9](#_Toc482987894)

[4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ 10](#_Toc482987895)

[ПРИЛОЖЕНИЕ 1 11](#_Toc482987896)

[ОПИСАНЕИ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ БИБЛИОТЕКИ RABIN.LIB 11](#_Toc482987897)

[ПРИЛОЖЕНИЕ 2 12](#_Toc482987898)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ, ПОЛЕЙ, СВОЙСТВ И КОНСТРУКТОРОВ ДОСТУПНЫХ ДЛЯ ИСПОЛЬЗОВАНИЯ ПРОГРАММИСТОМ 12](#_Toc482987899)

[ПРИЛОЖЕНИЕ 3 16](#_Toc482987900)

[ОПИСАНИЕ ФОРМАТА .publicKey 16](#_Toc482987901)

[ОПИСАНИЕ ФОРМАТА .privateKey 16](#_Toc482987902)

[ПРИЛОЖЕНИЕ 4 17](#_Toc482987903)

[СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ 17](#_Toc482987904)

# 1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ БИБЛИОТЕКИ

## **1.1. Назначение библиотеки**

Данная библиотека предназначена для кодирования и декодирования информации, представленной в виде: объекта класса String, объекта класса BigInteger, любого объекта наследника класса Stream. Реализуя алгоритм Криптосистемы Рабина.

## **1.2. Функции, выполняемые библиотекой**

Состав функций, выполняемых библиотекой Rabin.lib:

* Получение ключа
* Генерация закрытого ключа [Приложение 4] по указанной пользователем бинарной длине (пара простых чисел, дающих остаток 3 по модулю 4);
* Вычисление открытого ключа [Приложение 4] (произведение двух простых чисел, дающих остаток 3 по модулю 4);
* Возврат открытого ключа
  + Запись ключа в файл с расширением с расширением .publicKey [Приложение 3]
  + Возврат числа типа BigInteger
* Возврат закрытого ключа
  + Запись ключа в файл с расширением с расширением . privateKey [Приложение 3]
  + Возврат массива из двух чисел типа BigInteger
* Инициализация открытого ключа;
  + С помощью указания файла с расширением .publicKey [Приложение 3], в котором содержится информация об открытом ключе
  + С помощью числа типа BigInteger
* Инициализация закрытого ключа;
  + С помощью указания файла с расширением .privateKey [Приложение 3], в котором содержится информация об открытом ключе
  + С помощью двух простых чисел типа BigInteger, дающих остаток 3 по модулю 4
* Кодирование
* Кодирование числа типа BigInteger
* Кодирование информации, являющейся объектом класса наследника класса Stream
* Кодирование указанного файла и запись шифротекста в указанный файл с кодировкой RabinEncoding

Кодирование файласширениемлю 4ржится информация об открытом ключебованиям, перечисленным в документе класса

* Декодирование файла
* Декодирование числа типа BigInteger
* Декодирование информации, являющейся объектом класса наследника класса Stream
* Декодирование указанного файла и запись результата в указанный файл с кодировкой RabinEncoding
* Перевод объекта класса BigInteger в строку формата String в кодировке Unicode
* Перевод строки типа String в объект класса BigInteger в кодировке Unicode

## **1.3. Условия необходимые для выполнения библиотеки**

### **1.3.1. Требования к составу и параметрам технических средств**

Для надёжной и бесперебойной работы библиотеки требуется следующий состав технических средств:

* 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 1 ГГц [Приложение 4] или выше.
* 1 ГБ [Приложение 4] (для 32-разрядного процессора) или 2 ГБ (для 64-разрядного процессора) ОЗУ[Приложение 4].
* 16 ГБ (для 32-разрядной системы) или 20 ГБ (для 64-разрядной системы) свободного места на жестком диске.
* Графическое устройство DirectX 9 с драйвером WDDM [Приложение 4] 1.0 или более поздней версии.

### **1.3.2. Требования к программным средствам**

Для работы библиотеки необходим следующий состав программных средств:

* Операционная система Microsoft Windows 7 или более поздняя версия.
* Microsoft .NET Framework 4.5.2.

**2. ХАРАКТЕРИСТИКИ ПРОГРАММЫ**

**2.1. Временные характеристики**

Алгоритм кодирования файлов работает за , где n – длина числа

Алгоритм декодирования файлов работает за ), где n – длина числа, k = ((p + 1) / 4) % p, а p – одно из чисел закрытого ключа.

**2.2. Средства контроля правильности выполнения**

Библиотека реализует алгоритм Криптосистемы Рабина, способна кодировать/декодировать информацию/генерировать ключи как в основном потоке, так и в параллельном,овать информацию ть делегат рефлизующий интерфейс методы класса отсылая отчет о прогрессе.

# 3. ОБРАЩЕНИЕ К БИБЛИОТЕКЕ

Для использования библиотеки RabinLib необходимо подключить библиотеку System.Numerics и подключить саму библиотеку RabinLib к своей программе.

## **3.1. Задание ключа**

Перед началом кодировки или декодировки следует задать или сгенерировать новый ключ. Для этого нужно создать объект класса Rabin [Приложении 1]. Сделать это можно использовав конструкторы класса Rabin [Приложение 2].

### **3.1.1. Генерация ключей**

Генерация открытого и закрытого ключа [Приложение 4] выполняется путем создания объекта классаRabin, с помощью bitlength (int) – бинарная длина открытого ключа

Rabin rab = new Rabin(bitLength);

Остальные способы генерации ключа смотри в Приложении 2.

### **3.1.2. Задание существующих ключей**

Если требуется использовать уже созданные ключи, то требуется создать объект класса Rabin следующими способами:

* Создание объекта Rabin, задав открытый ключ: Rabin rab = new Rabin(BigInteger n); n – число, характеризующие открытый ключ;
* Создание объекта Rabin, задав закрытый ключ: Rabin rab = new Rabin(BigInteger p, BigInteger q); p и q – числа, характеризующие закрытый ключ;
* Создание объекта Rabin, передав путь к файлу на жестком диске компьютера, где записан открытый или закрытый ключ: Rabin rab = new Rabin(keyPath); keyPath – путь к файлу с ключем.

## **3.2. Кодирование**

После того как был выполнен п. 3.1. можно приступить к кодированию. Закодировать можно информацию, представленную в следующем формате:

* Число типа BigInteger
* Объект класса наследника класса Stream
* Файл хранящийся на диске компьютера

Чтобы закодировать строку формата String, нужно перевести ее в число BigInteger, воспользовавшись методом ToBigInteger [Приложение 2].

Способы выполнения, перечисленного выше функционала можно осуществить, используя методы класса Rabin [Приложение 2].

### **3.2.1. Кодирование числа**

Данная библиотека позволяет кодировать числа типа BigInteger. Для этого следует воспользоваться следующим методом: Encrypt(BigInteger C). Где C – кодируемое число. Если число С больше открытого ключа, то выбрасывается ошибка с сообщением потери данных C. В противном случае метод возвращает закодированное число.

### **3.2.2. Кодирование объекта наследника объекта Stream**

Для кодирования объекта наследника объекта Stream следует воспользоваться следующим методом: Encrypt(Stream read, Stream write, IProgress<ProgressInfo> progress).

Где read – кодируемый объект, write – объект, куда производится запись, progress – делегат, куда отсылается отчет о прогрессе.

**3.2.3. Кодирование файла**

Для кодирования файла, хранящегося на компьютере, следует воспользоваться следующим методом: Enrypt(string filename, string encryptPath, IProgress<ProgressInfo> progress = null). Где filename – путь к кодируемому файлу, encryptPath – путь к файлу, куда производится запись, progress – делегат, куда отсылается отчет о прогрессе. Перегрузки данного метода (EncryptAsync) смотри в Приложении 2.

## **3.3. Декодирование**

После того как был выполнен п. 3.1. можно приступить к декодированию. Раскодировать можно информацию, представленную в следующем формате:

* Число типа BigInteger
* Объект класса наследника класса Stream
* Файл хранящийся на компьютера

Чтобы раскодировать строку формата String, нужно перевести ее в число BigInteger, воспользовавшись методом ToString [Приложение 2].

Способы выполнения, перечисленного выше функционала можно осуществить, используя методы класса Rabin [Приложение 2].

### **3.3.1. Декодирование числа**

Данная библиотека позволяет декодировать числа типа BigInteger. Для этого следует воспользоваться следующим методом: Decrypt(BigInteger C). Где C – кодируемое число. Метод возвращает массив с четырьмя шифротекстами, один из которых верный.

### **3.3.2. Декодирование объекта наследника объекта Stream**

Для декодирования объекта наследника объекта Stream следует воспользоваться следующим методом: Decrypt(Stream read, Stream write, IProgress<ProgressInfo> progress).

Где read – декодируемый объект, write – объект, куда производится запись, progress – делегат, куда отсылается отчет о прогрессе.

**3.3.3. Декодирование файла**

Для декодирования файла, хранящегося на компьютере, следует воспользоваться следующим методом: Decrypt(string filename, string encryptPath, IProgress<ProgressInfo> progress = null). Где filename – путь к декодируемому файлу, encryptPath – путь к файлу, куда производится запись, progress – делегат, куда отсылается отчет о прогрессе. Перегрузки данного метода (DecryptAsync) смотри в Приложении 2.

## **3.4. Получение отчета о выполняемом процессе**

Большинство методов библиотеки кодирования/декодирования/генерации ключа RabinLib отсылают отчет о выполняемом процессе. Чтобы получать отчет о процессе, нужно в данные методы передать делегат реализующий интерфейс IProgress<ProgressInfo>. Информацию о классе ProgressInfo можно найти в Приложении 1 и Приложении 2.

# 4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Входными данными являютя:

* N (BigInteger) – открытый ключ
* PQ (BigInteger[]) – закрытый ключ
* С (BigInteger) – кодируемое число
* M (BigInteger[]) – массив с декодированными числами
* BitLength (int) – бинарная длина открытого ключа
* Str (string) – строка, переводимая в число BigInteger
* m (BigInteger) – число BigInteger, переводимое в стороку Str
* KeysPath (string) – путь на жестком диске компьютера, где находится папка с ключами
* publicKeyPath (string) – путь на жестком диске компьютера, где находится открытый ключ
* privateKeyPath (string) – путь на жестком диске компьютера, где находится закрытый ключ
* filename (string) – путь на жестком диске компьютера, где находится шифруемый и дешифруемый файл
* encryptPath (string) – путь на жестком диске компьютера, нуда записывать информацию
* read (Stream) – объект Stream, откуда надо считывать информацию
* write (Stream) – объект Stream, куда надо записывать информацию
* progress (IProgress <ProgressInfo>) – Прогресс процессов
* Tag (object) – данные об объекте

Входными данными являютя:

* N (BigInteger) – открытый ключ
* PQ (BigInteger[]) – закрытый ключ
* M (BigInteger[]) – массив с декодированными числами
* Str (string) – строка, полученная из числа BigInteger
* m (BigInteger) – число BigInteger, полученная из строки Str
* BitLength (int) – бинарная длина открытого ключа
* progress (IProgress <ProgressInfo>) – Прогресс процессов
* task (Task) – ссылка на асинхронный процесс
* Tag (object) – данные об объекте

# ПРИЛОЖЕНИЕ 1

## **ОПИСАНЕИ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ БИБЛИОТЕКИ RABIN.LIB**

|  |  |
| --- | --- |
| **Класс** | **Назначение** |
| Rabin.cs | Основной класс библиотеки RabinLib.dll. В этом классе реализована генерация открытых и закрытых ключей, кодирование и декодирование информации, представленной в виде: объекта класса String, объекта класса BigInteger, любого объекта наследника класса Stream. |
| RabinEncoding.cs | Класс, наследуемый от System.Text.Encoding.cs и реализующий кодировку, с помощью которой ведется запись информации в зашифрованные файлы и файлы с ключами. |
| ProgressInfo.cs | Класс, реализуемый для System.IProgress<T>, представляющий формат отчета, описывающего ход процесса. |

# ПРИЛОЖЕНИЕ 2

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ, ПОЛЕЙ, СВОЙСТВ И КОНСТРУКТОРОВ ДОСТУПНЫХ ДЛЯ ИСПОЛЬЗОВАНИЯ ПРОГРАММИСТОМ**

Rabin.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Аргументы* | | *Назначение* |
| Decrypt | public | BigInteger[] | BigInteger c | | Используя закрытый ключ, дешифрует сообщение c и возвращает массив с четырьмя шифротекстами, один из которых верный. |
| Encrypt | public | BigInteger | BigInteger m | | Используя открытый ключ, шифрует сообщение. |
| ToBigInteger | public | BigInteger | string str | | Переводит строку в число, используя кодировку Unicode. |
| ToString | public | string | BigInteger m | | Переводит число в строку, используя кодировку Unicode. |
| KeyToFile | public | void | String keysPath | | Сохраняет открытый и закрытый ключи в указанное место на диске. |
| Write  PublicKey | public | void | string publicKeyPath | | Записывает открытый ключ в указанный файл в кодировке RabinEncoding. |
| Write  PrivateKeys | public | void | string privateKeyPath | | Записывает закрытый ключ в указанный файл в кодировке RabinEncoding. |
| ReadPublicKey | public | static BigInteger | string publicKeyPath | | Считывает из указанного файла открытый ключ и возвращает его численное значение. |
| ReadPrivateKey | public | static BigInteger[] | string privateKeyPath | | Считывает из указанного файла закрытые ключи и  возвращает массив с их численными значениями. |
| Encrypt | public | void | string filename, string encryptPath, IProgress  <ProgressInfo> progress = null | | Используя открытый ключ, шифрует файл с именем filename и сохраняет зашифрованный файл в указанном месте. Отсылает информацию о процессе, если progress инициализирован пользователем. |
| EncryptAsync | public | async Task | string filename, string encryptPath, IProgress  <ProgressInfo> progress | | Используя открытый ключ, асинхронно шифрует файл с именем filename и сохраняет зашифрованный файл в указанном месте. Отсылает информацию о процессе. |
| Encrypt | public | void | Stream read, Stream write, IProgress  <ProgressInfo> progress | | Используя открытый ключ, кодирует информацию из первого объекта Stream и записывает ее во второй объект Stream. И отсылает информацию о процессе. |
| Decrypt | public | void | string filename, string decryptPath, IProgress  <ProgressInfo> progress = null | | Используя поля открытый ключ, дешифрует файл с именем filename и сохраняет зашифрованный файл в указанном месте. Отсылает информацию о процессе, если progress инициализирован пользователем. |
| DecryptAsync | public | async Task | string filename, string decryptPath, IProgress  <ProgressInfo> progress | | Используя закрытый ключ, асинхронно дешифрует файл с именем filename и сохраняет зашифрованный файл в указанном месте. Отсылает информацию о процессе, если progress инициализирован пользователем. |
| Decrypt | public | void | Stream read, Stream write, IProgress  <ProgressInfo> progress | | Используя закрытый ключ, декодирует сообщение из первого объекта Stream и записывает ее во второй объект Stream. И отсылает информацию о процессе. |
| **Конструкторы** | | | | | |
| *Имя* | *Мод. доступа* | *Аргументы* | | *Назначение* | |
| Rabin | public | int bitLength | | Инициализирует объект класса Rabin, генерируя открытые ключи и считая закрытый. | |
| Rabin | public | int bitLength, IProgress  <ProgressInfo> progress | | Инициализирует объект класса Rabin, генерируя открытые ключи и считая закрытый, отсылая информацию о прогрессе. | |
| Rabin | public | BigInteger n | | Инициализирует объект класса Rabin, используя значения открытого ключа. | |
| Rabin | public | BigInteger p, BigInteger q | | Инициализирует объект класса Rabin, используя значение закрытого ключа. | |
| Rabin | public | string keyPath | | Инициализирует объект класса Rabin, считывая из файла открытый или закрытые ключ, в зависимости от расширения файла. | |
| **Свойства** | | | | | |
| *Имя* | *Мод. доступа* | *Тип* | | *Назначение* | |
| Tag | public | object | | Возвращает или задает объект, содержащий данные об объекте. | |
| PQ | public | BigInteger[] | | Возвращает массив BigInteger, который содержит 2 числа, задающих закрытый ключ. | |
| N | public | BigInteger | | Возвращает открытый ключ. | |
| BitLength | public | int | | Возвращает бинарную длину открытого ключа. | |

ProgressInfo.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Конструкторы** | | | |
| *Имя* | *Мод. доступа* | *Аргументы* | *Назначение* |
| ProgressInfo | public | long newValue, long newMaximum | Инициализирует объект ProgressInfo, задавая новое положение и максимальное значение. |
| **Поля** | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Назначение* |
| Value | public | readonly long | Положение в данный момент. |
| maximum | public | readonly long | Максимальное значение. |

**ПРИЛОЖЕНИЕ 3**

**ОПИСАНИЕ ФОРМАТА .publicKey**

Файл формата .publicKey содержит открытый ключ записанный в кодировке RabinEncoding.

Преимуществами данной кодировки является уменьшение в 2 раза используемой памяти компьютера по сравнению со стандартной, UTF8Encoding. Кодировка также уменьшает читабельность информации другими программами, что повышает степень защиты от злоумышленников.

Корректно открыть файл .publicKey можно программой «Rabin».

**ОПИСАНИЕ ФОРМАТА .privateKey**

Файл формата .privateKey содержит закрытый ключ записанный в кодировке RabinEncoding.

Преимуществами данной кодировки является уменьшение в 2 раза используемой памяти компьютера по сравнению со стандартной, UTF8Encoding. Кодировка также уменьшает читабельность информации другими программами, что повышает степень защиты от злоумышленников.

Корректно открыть файл .privateKey можно программой «Rabin».

# ПРИЛОЖЕНИЕ 4

## **СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ**

ГБ – Согласно ГОСТ 8.417—2002 термин гигабайт с обозначением «ГБ» равен 109 = 1 000 000 000 байт.

Гц – Единица частоты периодических процессов ( 1 Гц = 1 с-1 ).

ОЗУ – Оперативно запоминающее устройство

WDDM – это архитектура графических драйверов для видеокарты под управлением Microsoft Windows.

Закрытый ключ – пара простых чисел P и Q, дающих остаток 3 при делении на 4. С помощью закрытого ключа производится дешифровка сообщения.

Открытый ключ – число N, численно равное произведению пары простых чисел P и Q. С помощью открытого ключа производится шифрования сообщения.

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированх |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |