**АННОТАЦИЯ**

Техническое задание – это основной документ, оговаривающий набор требований и порядок создания программного продукта, в соответствии с которым производится разработка программы, ее тестирование и приемка.

Настоящее Техническое задание на разработку «Программы для построения графов данных и поиска сообществ в них» содержит следующие разделы: «Введение», «Основание для разработки», «Назначение разработки», «Требования к программе», «Требования к программным документам», «Технико-экономические показатели», «Стадии и этапы разработки», «Порядок контроля и приемки» и приложения [7].

В разделе «Введение» указано наименование и краткая характеристика области применения «Программы для построения графов данных и поиска сообществ в них».

В разделе «Основания для разработки» указан документ на основании, которого ведется разработка и наименование темы разработки.

В разделе «Назначение разработки» указано функциональное и эксплуатационное назначение программного продукта.

Раздел «Требования к программе» содержит основные требования к функциональным характеристикам, к надежности, к условиям эксплуатации, к составу и параметрам технических средств, к информационной и программной совместимости, к маркировке и упаковке, к транспортировке и хранению, а также специальные требования.

Раздел «Требования к программным документам» содержит предварительный состав программной документации и специальные требования к ней.

Раздел «Технико-экономические показатели» содержит ориентировочную экономическую эффективность, предполагаемую годовую потребность, экономические преимущества разработки «Программа для построения графов данных и поиска сообществ в них».

Раздел «Стадии и этапы разработки» содержит стадии разработки, этапы и содержание работ.

В разделе «Порядок контроля и приемки» указаны общие требования к приемке работы.

Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.102-77 Стадии разработки [2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению [7].

Изменения к данному Техническому заданию оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

СОДЕРЖАНИЕ

[1. Введение 5](#_Toc385162099)

[**1.1. Наименование программы 5**](#_Toc385162100)

[**1.2. Краткая характеристика области применения 5**](#_Toc385162101)

[2. Основания для разработки 6](#_Toc385162102)

[**2.1. Документы, на основании которых ведется разработка 6**](#_Toc385162103)

[**2.2. Наименование темы разработки 6**](#_Toc385162104)

[3. Назначение разработки 7](#_Toc385162105)

[**3.1. Функциональное назначение 7**](#_Toc385162106)

[**3.2. Эксплуатационное назначение 7**](#_Toc385162107)

[4. Требования к программе 8](#_Toc385162108)

[**4.1. Требования к функциональным характеристикам 8**](#_Toc385162109)

[**4.1.1. Требования к составу выполняемых функций 8**](#_Toc385162110)

[**4.1.2. Требования к организации входных данных 8**](#_Toc385162112)

[**4.1.3. Требования к организации выходных данных 8**](#_Toc385162113)

[**4.1.4. Требования к временным характеристикам 8**](#_Toc385162114)

[**4.2. Требования к интерфейсу 8**](#_Toc385162115)

[**4.3. Требования к надежности 8**](#_Toc385162116)

[**4.3.1. Требования к обеспечению надежного (устойчивого) функционирования программы 8**](#_Toc385162117)

[**4.3.2. Время восстановления после отказа 9**](#_Toc385162118)

[**4.3.3. Отказы из-за некорректных действий оператора 9**](#_Toc385162119)

[**4.4. Условия эксплуатации 9**](#_Toc385162120)

[**4.4.1. Климатические условия эксплуатации 9**](#_Toc385162121)

[**4.4.2. Требования к видам обслуживания 9**](#_Toc385162122)

[**4.4.3. Требования к численности и квалификации персонала 10**](#_Toc385162123)

[**4.5. Требования к составу и параметрам технических средств 10**](#_Toc385162124)

[**4.6. Требования к информационной и программной совместимости 10**](#_Toc385162125)

[**4.6.1. Требования к информационным структурам и методам решения 10**](#_Toc385162126)

[**4.6.2. Требования к программным средствам, используемым программой. 10**](#_Toc385162127)

[**4.6.3. Требования к исходным кодам и языкам программирования 10**](#_Toc385162128)

[**4.6.4. Требования к защите информации и программы 10**](#_Toc385162129)

[**4.7. Требования к маркировке и упаковке 11**](#_Toc385162130)

[**4.8. Требования к транспортировке и хранению 11**](#_Toc385162131)

[**4.8.1. Требования к хранению и транспортировке компакт-дисков (CD) 11**](#_Toc385162132)

[**4.8.2. Требования к хранению и транспортировке программных документов, предоставляемых в печатном виде. 11**](#_Toc385162133)

[**4.9. Специальные требования 12**](#_Toc385162134)

[5. Требования к программной документации 13](#_Toc385162135)

[**5.1. Предварительный состав программной документации 13**](#_Toc385162136)

[**5.2. Специальные требования к программной документации 13**](#_Toc385162137)

[6. Технико-экономические показатели 14](#_Toc385162138)

[**6.1. Ориентировочная экономическая эффективность 14**](#_Toc385162139)

[**6.2. Предполагаемая потребность 14**](#_Toc385162140)

[**6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 14**](#_Toc385162141)

[7. Стадии и этапы разработки 15](#_Toc385162142)

[8. Порядок контроля и приемки 16](#_Toc385162143)

[**8.1. Виды испытаний 16**](#_Toc385162144)

[**8.2. Общие требования к приемке работы 16**](#_Toc385162145)

[Приложение 1. Список используемой литературы 18](#_Toc385162148)

1. **ВВЕДЕНИЕ**
   1. **Наименование программы**

Наименование программы – «Программа для построения графов данных и поиска сообществ в них».

* 1. **Краткая характеристика области применения**

«Программа для построения графов данных и поиска сообществ в них» – программа, позволяющая построить по данным различные соответствующие им графы, а также реализующая поиск сообществ в взвешенном или не взвешенном неориентированном графе.

Сообществами называются такие подмножества вершин, у которых много рёбер внутри подмножества и мало – с остальными вершинами[11, 12]. Существуют более строгие определения сообщества, но они не общеприняты. Поиск сообществ в сети важен, поскольку с его помощью можно изучить структуру сети, выявив в ней основные части и взаимодействия между ними.

Помимо рёбер на основе явно заданных связей между вершинами в сложной сети можно строить рёбра на основе метаданных, ассоциированных с вершинами, соединяя вершины со схожими метаданными. Такой граф называется графом данных (proximity graph). Существует несколько методов построения графа данных[13] (relative neigborhood graph, gabriel graph и т.д.), некоторые из которых планируется реализовать.

Таким образом, данная программа может применяться при исследовании различных наборов данных для выявления их структуры -- выделения основных частей данных и связей между ними.

1. **ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ**
   1. **Документы, на основании которых ведется разработка**

Основанием для разработки является Приказ декана факультета компьютерных наук И.В. Аржанцева от 12.12.2018 № 2.3-02/1212-02 "Об утверждении тем, руководителей и консультантов курсовых работ.

* 1. **Наименование темы разработки**

Программа выполнена в рамках темы курсового проекта — «Сложные сети: граф связей и граф данных», в соответствии с учебным планом подготовки бакалавров по направлению «Прикладная математика и информатика».

1. **НАЗНАЧЕНИЕ РАЗРАБОТКИ**
   1. **Функциональное назначение**

Функциональным назначением программы является построение графов по данным и поиск сообществ в полученных графах.

* 1. **Эксплуатационное назначение**

Данная программа может применяться при исследовании различных наборов данных для их изучения и для выявления их структуры.

1. **ТРЕБОВАНИЯ К ПРОГРАММЕ**
   1. **Требования к функциональным характеристикам**
      1. **Требования к составу выполняемых функций**

Программа должна обеспечивать возможность построения следующих графов данных:

1. Граф K ближайших соседей.

Каждая вершина соединяется с её K ближайшими соседями.

2. Граф пар вершин на расстоянии меньше заданного.

При этом должна поддерживаться возможность не только явно задавать это расстояние, но и автоматически находить его, передавая в функцию желаемое количество ребёр в построенном графе.

3. RNG (relative neighbourhood graph)

Две точки в пространстве, соответствующие объектам из набора данных, соединяются рёбром, если в пересечении двух гиперсфер с центрами в каждой из точек и радиусом, равным расстоянию между ними, нет других точек из набора данных.

4. Граф Габриеля.

Две точки в пространстве соединяются рёбром, если в пересечении двух гиперсфер с центрами в каждой из точек и радиусом, равным расстоянию между ними, нет других точек из набора данных.

5. Граф сфер влияния.

Для каждой точки строится гиперсфера с радиусом, равным расстоянию до её ближайшего соседа. Далее пара точек соединяется ребром, если соответствующие им гиперсферы пересекаются.

Предполагается реализация следующих алгоритмов поиска сообществ:

1. Label propagation, синхронная и асинхронная модификации.[14]

2. Алгоритм CNM.[15]

* + 1. **Требования к организации входных данных**

Входными данными для реализуемых функций являются контейнеры стандартной библиотеки языка С++.

Входными данными для интерфейса на языке Python3 являются контейнеры стандартной библиотеки языка Python3, для представления графов – класс граф из библиотеки networkx. Поскольку в Python3 используется «утиная типизация», могут использоваться любые совместимые с вышеперечисленными типы данных.

* + 1. **Требования к организации выходных данных**

Выходными данными для реализуемых функций являются контейнеры стандартной библиотеки языка С++.

Выходными данными для интерфейса на языке Python3 являются контейнеры стандартной библиотеки языка Python3, для представления графов – класс Graph из библиотеки networkx. Поскольку в Python3 используется «утиная типизация», могут использоваться любые совместимые с вышеперечисленными типы данных.

* + 1. **Требования к временным характеристикам**

Реализованные алгоритмы должны быть по возможности масштабируемыми для возможности обработки данных большого размера.

* 1. **Требования к интерфейсу**

Данный программный продукт должен быть реализован как библиотека на языке С++, использующая входные и выходные данные в формате контейнеров стандартной библиотеки языка С++, либо в формате определённых в библиотеке классов.

Кроме того должен быть написан интерфейс библиотеки для языка Python3.

* 1. **Требования к надежности**
     1. **Требования к обеспечению надежного (устойчивого) функционирования программы**

Для устойчивой работы программы необходимо соблюдать ряд организационно-технических мер:

1. обеспечить бесперебойное питание технических устройств;
2. обеспечить высокую защиту технических устройств для работы программы от воздействия шпионских программ, троянских программ, программ-шуток и других видов вредоносного программного обеспечения;
3. обеспечить регулярную проверку оборудования и программного обеспечения на наличие сбоев и неполадок;
4. обеспечить использование лицензионного программного обеспечения.
   * 1. **Время** **восстановления после** **отказа**

Если отказ был вызван какими-либо внешними факторами, например, сбоем электропитания, и при этом не произошел непоправимый сбой операционной системы, то время восстановления не должно превышать времени, требующегося на перезагрузку операционной системы и запуск программы.

Если отказ был вызван неисправностью технических средств или непоправимым сбоем операционной системы, то время восстановления не должно превышать времени, необходимого для устранения неисправностей технических и программных средств.

* + 1. **Отказы из-за некорректных действий оператора**

Отказ программы возможен также вследствие некорректных действий пользователя при пользовании операционной системой. Для предотвращения случаев отказа программы по причине сбоев при пользовании операционной системой следует провести предварительный инструктаж конечного пользователя и обеспечить работу конечного пользователя без предоставления ему прав администратора.

Для недопущения отказа программы вследствие некорректного ввода данных следует проверять их корректность.

* 1. **Условия эксплуатации**
     1. **Климатические условия эксплуатации**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к персональным компьютерам в части условий их эксплуатации.

Персональный компьютер предназначен для работы в закрытом отапливаемом помещении со стабильными климатическими условиями категории 4.1 согласно ГОСТ 15150-69[10].

* + 1. **Требования к видам обслуживания**

На персональном компьютере, где производится эксплуатация программы необходимо обеспечить регулярные проверки оборудования и программного обеспечения на наличие сбоев и неполадок. Обеспечить защиту персонального компьютера от воздействия шпионских программ, программ-шуток, троянских программ и других видов вирусов.

Если произошел какой-либо непредвиденный сбой в программе, то пользователю для устранения текущих неполадок рекомендуется написать разработчику на адрес электронной почты указанный в разделе «Справка» и сообщить обо всех замеченных сбоях. Разработчик в свою очередь обязан принять меры по устранению неполадок и выслать пользователю исправленную версию программного продукта.

* + 1. **Требования к численности и квалификации персонала**

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 1 штатной единицы:

1. конечный пользователь – оператор ЭВМ.

Оператор ЭВМ должен быть способен подключить и использовать библиотеку на языке С++, либо на языке Python3.

* 1. **Требования к составу и параметрам технических средств**

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

1) Компьютер. Требования к размеру оперативной памяти и производительности компьютера зависят от размера обрабатываемых данных.

* 1. **Требования к информационной и программной совместимости**
     1. **Требования к информационным структурам и методам решения**

Требования к методам решения не предъявляются.

* + 1. **Требования к программным средствам, используемым программой.**

Для работы программы необходим следующий состав программных средств:

1) Компилятор языка С++ (при использовании библиотеки в языке С++)

2) Интерпретатор языка Python3 (при использовании интерфейса для Python3)

3) Среда разработки или текстовый редактор.

* + 1. **Требования к** **исходным** **кодам и** **языкам** **программирования**

Программа должна быть написана на языке программирования C++ . Интерфейс для Python3 должен быть написан на языке Python3.

* + 1. **Требования к защите информации и программы**

Требования к защите информации и программы не предъявляются.

* 1. **Требования к маркировке и упаковке**

Особых требований нет.

* 1. **Требования к транспортировке и хранению**
     1. **Требования к хранению и транспортировке программы**

Программное изделие может храниться и транспортироваться на флеш носителе и в облачном хранилище. Может храниться и распространяться в виде исходного кода, либо разделяемой библиотки (в формате .so или .dll).

Интерфейс для языка Python3 хранится в виде модуля с кодом на этом языке.

Исходный код должен быть доступен по адресу:

https://github.com/alexey9177950/data-graph

* + 1. **Требования к хранению и транспортировке программных документов, предоставляемых в печатном виде.**

Требования к транспортировке и хранению программных документов являются стандартными и должны соответствовать общим требованиям хранения и транспортировки печатной продукции:

1. В помещении для хранения печатной продукции допустимы температура воздуха от 10°С до 30°С и относительная влажность воздуха от 30% до 60%.
2. Документацию хранят и используют на расстоянии не менее 0.5 от источников тепла и влаги. Не допускается хранение печатной продукции в помещениях, где находятся агрессивные агенты – растворители, спирт, бензин.
3. Не допускается попадание на документацию агрессивных агентов.
4. Транспортировка производится в специальных контейнерах с применением мер по предотвращению деформации документов внутри контейнеров, а также проникновения влаги, вредных газов, пыли, солнечных лучей и образованию конденсата внутри контейнеров.
5. Программные документы, предоставляемые в печатном виде должны соответствовать общим правилам учета и хранения программных документов, предусмотренных стандартами Единой системы программной документации и соответствовать требованиям ГОСТ 19.602-78.

* 1. **Специальные требования**

Специальные требования к данной программе не предъявляются.

1. **ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ**
   1. **Предварительный состав программной документации**
2. «Программа для построения графов данных и поиска сообществ в них». Техническое задание (ГОСТ 19.201-78);
3. «Программа для построения графов данных и поиска сообществ в них». Текст программы (ГОСТ 19.401-78);
4. «Программа для построения графов данных и поиска сообществ в них». Пояснительная записка (ГОСТ 19.404-79);
   1. **Специальные требования к программной документации**
5. Все документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 [6] и ГОСТ к этому виду документа (см. п. 5.1.).
6. Пояснительная записка должна быть загружена в систему Антиплагиат через ЛМС НИУ ВШЭ. Лист, подтверждающий загрузку пояснительной записки, сдается в учебный офис вместе со всеми материалами не позже, чем за день до защиты курсовой работы.
7. Вся документация сдается в печатном виде, при этом она должна быть обязательно подписана академическим руководителем образовательной программы «Прикладная математика и информатика», руководителем разработки и исполнителем перед сдачей курсовой работы в учебный офис не позже одного дня до защиты.
8. Вся документация и программа также сдается в электронном виде в формате .pdf или .docx. в архиве формата .rar или .zip.
9. Все документы перед защитой курсовой работы должны быть загружены в информационно-образовательную среду НИУ ВШЭ LMS (Learning management system) в личном кабинете, дисциплина - «Курсовая работа», одним архивом (см. п.3)
10. **ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**
    1. **Ориентировочная экономическая эффективность.**

В рамках данной работы расчет экономической эффективности не предусмотрен.

**6.2.** [**Предполагаемая потребност**](#_Toc385162140)**ь.**

Потребность в программе может возникнут при исследовании структуры различных наборов данных.

[**6.3.**](#_Toc385162141) **Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами**Error: Reference source not found

Поиск в сети интернет не выявил аналогов программы на момент разработки.

1. **СТАДИИ И ЭТАПЫ РАЗРАБОТКИ**

Стадии и этапы разработки были выявлены с учетом ГОСТ 19.102-77 [2]:

|  |  |  |
| --- | --- | --- |
| **Стадии разработки** | **Этапы работ** | **Содержание работ** |
| 1. **Техническое задание** | Обоснование необходимости разработки программы | Постановка задачи |
| Сбор исходных материалов |
| Научно-исследовательские работы | Определение структуры входных и выходных данных. |
| Определение требований к техническим средствам. |
| Обоснование принципиальной возможности решения поставленной задачи |
| Разработка и утверждение технического задания | Определение требований к программе. |
| Определение стадий, этапов и сроков разработки программы и документации на неё. |
| Согласование и утверждение технического задания. |
| **2. Пояснительная записка.** | Разработка пояснительной записки. | Разработка пояснительной записки. |
| **3. Рабочий проект** | Разработка программы | Программирование и отладка программы. |
| Разработка программной документации | Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 [1]. |
| Испытания программы | Разработка, согласование и утверждение порядка и методики испытаний. |
| **4. Внедрение** | Подготовка и защита программного продукта. | Подготовка программы и программной документации для презентации и защиты. |
| Утверждение дня защиты программы. |
| Презентация программного продукта. |
| Передача программы и программной документации в архив НИУ ВШЭ. |

1. **ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ**
   1. **Виды испытаний**

Производится проверка корректного выполнения программой заложенных в нее функций, т.е. осуществляется функциональное тестирование программы.

Функциональное тестирование осуществляется в соответствии с документом «Программа для построения графов данных и поиска сообществ в них». Программа и методика испытаний (ГОСТ 19.301-79), в котором указывают [18]:

1. перечень функций программы, выделенных в программе для испытаний, и перечень требований которым должны соответствовать эти функции (со ссылкой на пункт 4.1.1. настоящего технического задания);
2. перечень необходимой документации и требования к ней (со ссылкой на пункт 5 настоящего технического задания);
3. методы испытаний и обработки информации;
4. технические средства и порядок проведения испытаний;

Сроки проведения испытаний обсуждаются дополнительно.

* 1. **Общие требования к приемке работы**

Прием программного продукта происходит при полной работоспособности программы при различных входных данных, при выполнении указанных в пункте 4.1.1 настоящего документа функций, при выполнении требований указанных в пункте 4.2. настоящего документа и при наличии полной документации к программе, указанной в пункте 5.1, выполненной в соответствии со специальными требования указанными в пункте 5.2 настоящего технического задания.

**ПРИЛОЖЕНИЕ 1**

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
11. Santo Fortunato, Darko Hric, ”Community detection in networks: A user guide”, 2016.
12. Santo Fortunato, ”Community detection in graphs”, 2009.

13) Matthias Beck, ”Computational Discrete Geometry”

14) Usha Nandini Raghavan, Reka Albert, Soundar Kumara , ”Near linear time algorithm to detect community structures in large-scale networks”, 2009

15) Aaron Clauset, M. E. J. Newman,and Cristopher Moore, ”Finding community structure in very large networks”, 2004.

# 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 |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |