**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| СОГЛАСОВАНО  Научный руководитель, профессор департамента «Программной инженерии», доктор технических наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_А.И. Легалов  «\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  старший преподаватель департамента программной инженерии  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.А. Павлочев  «\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Кросс-компилятор с языка программирования Small C в ассемблер процессора**  **RISC-V, для эмулятора RARS**  **Техническое задание**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.04.01-01 ТЗ 01-1-ЛУ** | | | | | | |
|  | |  | | | | |
| Исполнитель  студент группы БПИ-229  \_\_\_\_\_\_\_\_\_ / Р.С. Громов /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | | |
|  | | | | | | |
|  | | | | |  | |

**Москва 2024**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.04.01-01 ТЗ 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Кросс-компилятор с языка программирования Small C в ассемблер**  **процессора RISC-V, для эмулятора RARS**  **Техническое задание**  **RU.17701729.10.03-01 ТЗ 01-1**  **Листов 20** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | | |
| **Москва 24** | | | | **24** |

# **АННОТАЦИЯ**

Настоящее Техническое задание на разработку «Кросс-компилятора с языка программирования Small C в ассемблер процессора RISC-V, для эмулятора RARS» 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.

В разделе «Введение» указано наименование и краткая характеристика области применения программы.

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 программы.

Раздел «Стадии и этапы разработки» содержат 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]](#_1v1yuxt);
2. ГОСТ 19.102-77 Стадии разработки [[2]](#_1v1yuxt);
3. ГОСТ 19.103-77 Обозначения программ и программных документов [[3]](#_1v1yuxt);
4. ГОСТ 19.104-78 Основные надписи [[4]](#_1v1yuxt);
5. ГОСТ 19.105-78 Общие требования к программным документам [[5]](#_1v1yuxt);
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [[6]](#_1v1yuxt);
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению [[7]](#_1v1yuxt).

Изменения к данному Техническому заданию оформляются согласно ГОСТ 19.603-78 [[8]](#_1v1yuxt), ГОСТ 19.604-78 [[9]](#_1v1yuxt).

**СОДЕРЖАНИЕ**

[**АННОТАЦИЯ** 3](#_Toc158849246)

[**1.** **ВВЕДЕНИЕ** 6](#_Toc158849247)

[**1.1.** **Наименование программы** 6](#_Toc158849248)

[**1.2.** **Краткая характеристика и область назначения** 6](#_Toc158849249)

[**2.** **ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ** 7](#_Toc158849250)

[**2.1.** **Документы, на основании которых ведется разработка** 7](#_Toc158849251)

[**2.2.** **Наименование темы разработки** 7](#_Toc158849252)

[**3.** **НАЗНАЧЕНИЕ РАЗРАБОТКИ** 8](#_Toc158849253)

[**3.1.** **Функциональное назначение** 8](#_Toc158849254)

[**3.2.** **Эксплуатационное назначение** 8](#_Toc158849255)

[**4.** **ТРЕБОВАНИЯ К ПРОГРАММЕ** 9](#_Toc158849256)

[**4.1.** **Требования к функциональным характеристикам** 9](#_Toc158849257)

[**4.1.1.** **Требования к составу выполняемых функций** 9](#_Toc158849258)

[**4.2.** **Требования к временным характеристикам** 10](#_Toc158849259)

[**4.3.** **Требования к интерфейсу** 10](#_Toc158849260)

[**4.4.** **Требования к надежности** 10](#_Toc158849261)

[**4.5.** **Условия эксплуатации** 11](#_Toc158849262)

[**4.6.** **Требования к составу и параметрам программных средств** 11](#_Toc158849263)

[**4.7.** **Требования к информационной и программной совместимости** 11](#_Toc158849264)

[**4.8.** **Требования к маркировке и упаковке** 11](#_Toc158849265)

[**4.9.** **Требования к транспортировке и хранению** 11](#_Toc158849266)

[**5.** **ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ** 12](#_Toc158849267)

[**5.1.** **Состав программной документации** 12](#_Toc158849268)

[**5.2.** **Специальные требования к программной документации** 12](#_Toc158849269)

[**6.** **ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ** 13](#_Toc158849270)

[**6.1.** **Ориентировочная экономическая эффективность**  13](#_Toc158849271)

[**6.2.** **Предполагаемая потребность** 13](#_Toc158849272)

[**7.** **СТАДИИ И ЭТАПЫ РАЗРАБОТКИ** 14](#_Toc158849273)

[**8.** **ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ** 17](#_Toc158849274)

[**8.1** **Виды испытаний** 17](#_Toc158849275)

[**8.2** **Общие требование к приёмке работы** 17](#_Toc158849276)

[**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ** 18](#_Toc158849277)

[**ГЛОССАРИЙ** 19](#_Toc158849278)

# **ВВЕДЕНИЕ**

## **Наименование программы**

Наименование программы — «Кросс-компилятор с языка программирования Small C в ассемблер процессора RISC-V для эмулятора RARS».

Наименование программы на английском языке — «Cross-compiler from the Small C Programming Language into the Assembler of the RISC-V Processor, for the RARS Emulator».

Краткое наименование программы — «RISC-V Cross-compiler».

## **1.2. Краткая характеристика и область назначения**

«RISC-V Cross-compiler» - приложение, которое транслирует высокоуровневый исходный код программы, написанной на языке программирования Small C, в низкоуровневый код на ассемблере процессора RISC-V для эмулятора RARS путём проведения препроцессинга, лексического, синтаксического, семантического анализа и генерации промежуточного кода.

Основная цель создания данного компилятора - помощь студентам «Программной инженерии» в более углубленном изучении дисциплины «Архитектура вычислительных систем», которая проводится с использованием ассемблера RISC-V и эмулятора RARS.

С помощью «RISC-V Cross-compiler» обучающиеся смогут:

1. Быстро и эффективно транслировать знакомый им высокоуровневый простой код языка Small C в более сложный ассемблер архитектуры RISC-V, который им предстоит изучать
2. Повысить уровень своих знаний по дисциплине «Архитектура вычислительных систем» путём исследования полученных ассемблерных инструкций, их связи с высокоуровневым кодом, а также их автоматический запуск на эмуляторе RARS
3. Разобраться в технологии трансляции высокоуровнего кода в низкоуровневый
4. Легче справляться с выполнением практических заданий на языке ассемблер процессора RISC-V

# **2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ**

## **2.1. Документы, на основании которых ведется разработка**

Программа выполнена в рамках задания на курсовую работу в соответствии с учебным планом подготовки бакалавров (НИУ ВШЭ, факультет компьютерных наук) по направлению 09.03.04 «Программная инженерия» и утверждённой академическим руководителем темы курсового проекта. Приказ декана факультета компьютерных наук И. В. Аржанцева "Об утверждении тем, руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета компьютерных наук"

## **2.2. Наименование темы разработки**

Наименование темы разработки – «Кросс-компилятор с языка программирования Small C в ассемблер процессора RISC-V для эмулятора RARS».

Условное обозначение темы разработки – «RISC-V Cross-compiler».

# **НАЗНАЧЕНИЕ РАЗРАБОТКИ**

## **Функциональное назначение**

Функциональное назначение «RISC-V Cross-compiler» представимо в виде списка основных функциональных групп приложения:

1. Выбор исходных файлов программы, написанной на языке Small C, а также их валидация для её дальнейшей компиляции.
2. Препроцессинг: получение отформатированной копии программы путём удаления лишних символов и комментариев и передача её лексическому анализатору
3. Компиляция: получение ассемблерного листинга исходной программы в ходе выполнения лексического, синтаксического, семантического анализа и генерации промежуточного кода
4. Получение информации о синтаксических и компиляционных ошибках, полученных в ходе компиляции, а также предупреждений о возможных ошибках в ходе выполнения ассемблерных инструкций либо на эмуляторе RARS, либо на компьютере архитектуры RISC-V
5. Генерации промежуточного кода в ассемблерные инструкции архитектуры RISC-V
6. Запуск ассемблерного кода на эмуляторе RARS

## **Эксплуатационное назначение**

Приложение «RISC-V Cross-compiler» представляет собой полноценный кроссплатформенный компилятор для трансляции высокоуровнего кода в ассемблерные инструкции с

Целевой аудиторией «RISC-V Cross-compiler» являются студенты «Программной инженерии», которые желают повысить уровень своих знаний по дисциплине «Архитектура вычислительных систем», а также хотят легче справляться с выполнением практических заданий по языку ассемблера процессора RISC-V

# **ТРЕБОВАНИЯ К ПРОГРАММЕ**

## **Требования к функциональным характеристикам**

### **Требования к составу выполняемых функций**

Программа должна обеспечивать возможность выполнения описанных ниже функций:

1. Выбор исходного файла для дальнейшей компиляции
   1. Компилятор должен запускаться с использованием интерфейса командной строки
   2. Ввод в консоль Формат командной строки должен включать в себя ввод абсолютного или относительного пути к исходному файлу с указанием его имени и расширения
   3. Валидация корректности введённого пути, имени файла и расширения
   4. Вывод информации об ошибках, которые могут возникнуть во время валидации или чтения найденного файла
2. Выполнение препроцессинга
   1. Удаление лишних пробельных символов и всех символов табуляции
   2. Удаление однострочных и многострочных комментариев
   3. Сохранение отформатированной копии программы в ту же папку, в которой находился исходный код программы
3. Выполнение компиляции
   1. Выполнение лексического анализа
      1. Разбиение отформатированного текста исходной программы на последовательность токенов
      2. Классификация полученных токенов
      3. Проверка каждого токена на валидность в соответствии с языком программирования Small C
   2. Выполнение синтаксического анализа
      1. Чтение потока токенов для обнаружения лексических и синтиаксических ошибок
      2. Восстановление работы анализатора при обнаружении ошибки
      3. Вывод информации об ошибке с указанием её типа, описания и расположения в коде исходной программы
      4. Построение абстрактного синтаксического дерева (АСТ) при отсутствии ошибок
   3. Выполнение семантического анализа
      1. Обход абстрактного синтаксического дерева для обнаружения семантических ошибок. Выполнение проверки типов
      2. Генерация промежуточного кода
   4. Генерация кода с учётом особенности архитектуры RISC-V
4. Сохранение текста исходной программы на языке ассемблера процессора RISC-V в папку с исходной программы языка Small C
5. Сформированная программа на ассемблере симулятора RARS должна являться исходными данными для этого симулятора и выполняться в его среде

## **Требования к временным характеристикам**

Приложение «RISC-V Cross-compiler» должно выполнять компиляцию исходной программы за время, не превышающее двух секунд. При большом объёме файла исходной программы или большом количестве используемых библиотек допускается превышение лимита времени. При задержке, которая превышает две секунды, необходимо уведомить об этом пользователя с соответствующим предупреждением о скорости работы приложения.

## **Требования к интерфейсу**

К десктопной версии выдвигаются следующие требования: интерфейс приложения должен представлять из себя консольное окно.

## **Требования к надежности**

Особых условий к надёжности программного продукта не предполагается, «RISC-V Cross-compiler» должен сохранять полную работоспособность при большом объёме исходного файла, компилятор не должен зависать, а также завершаться аварийно.

## **Условия эксплуатации**

Особых условий эксплуатации не выдвигается, использование программы должно быть интуитивно понятным для начинающего программиста.

## **Требования к составу и параметрам программных средств**

Программа должна работать на любом персональном компьютере, обеспечивающем функционирование симулятора RARS и трансляцию разрабатываемого компилятора с использованием современных компиляторов языка C

## **Требования к информационной и программной совместимости**

Никаких особых требований к информационной и программной совместимости не выдвигается. Возможно использование любых средств разработки.

## **Требования к маркировке и упаковке**

Специальных требований не предполагается. Исходники приложения, а также его сборки находятся в онлайн репозитории github.com. К приложению отдельно поставляется его документация в формате pdf

## **Требования к транспортировке и хранению**

Исходники приложения, а также его сборки находятся в онлайн репозитории github.com К документации особенных требований не предполагается.

# **ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ**

## **Состав программной документации**

* «RISC-V Cross-compiler». Техническое задание (ГОСТ 19.201-78);
* «RISC-V Cross-compiler». Программа и методика испытаний (ГОСТ 19.30178);
* «RISC-V Cross-compiler». Текст программы (ГОСТ 19.40178);
* «RISC-V Cross-compiler». Пояснительная записка (ГОСТ 19.40479);
* «RISC-V Cross-compiler». Руководство оператора (ГОСТ 19.50579).

## **Специальные требования к программной документации**

* Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. п. 5.1.);
* Пояснительная записка должна быть загружена в систему Антиплагиат «НИУ ВШЭ»;
* Техническое задание и пояснительная записка, титульные листы остальных документов должны быть подписаны руководителем разработки и исполнителем;
* Документация и программа сдается в электронном виде в формате .рdf или .dосх в архиве формата .zip или .rar.
* За три дня до защиты курсового проекта все следующие материалы:

1. Программный проект;
2. Программная документация;
3. Исполняемые файлы
4. Отзыв руководителя
5. Отчёт системы Антиплагиат

должны быть загружены через систему SmartLMS в дисциплину «Курсовой проект» одниv или несколькими архивами формата .rar или .zip

# **ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

## **Ориентировочная экономическая эффективность** В рамках данной работы расчёт экономической эффективности не предусмотрен

## **Предполагаемая потребность**

Эта разработка будет интересна людям, обучающимся на «Программной инженерии»

# **СТАДИИ И ЭТАПЫ РАЗРАБОТКИ**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Стадии разработки** | **Этапы работ** | **Содержание работ** | **Исполнители работ** | **Сроки выполнения** |
| 1. Техническое задание | Подготовительные работы | * Постановка задачи * Сбор исходных теоретических и статистических материалов. * Обоснование возможностей решения поставленной задачи. * Определение структуры входных и выходных данных. * Предварительный выбор методов решения задач. | Р. С. Громов | 15.10.2023 –15.11.2023 |
| Разработка и утверждение технического здания | * Определение требований к программе. * Определение требований к техническим средствам. * Определение стадий. Этапов и сроков разработки программы и документации на неё. * Согласование и утверждение технического задания. | Р. С. Громов | 16.11.2023 – 15.01.2024 |
| 1. Прототип проекта | Разработка архитектуры | Обсуждение и дизайн архитектуры проекта | Р. С. Громов | 16.01.2024 – 30.01.2024 |
| Разработка прототипа | Создание рабочего прототипа для одной из платформ |
| Тестирование прототипа | Проведение испытаний для определения качества спроектированного прототипа  Изменение архитектуры при неудовлетворительном результате |
| 1. Рабочий проект | Разработка программы | Программирование и отладка программы | Р. С. Громов | 01.02.2024–15.03.2024 |
| Разработка программной документации | Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 | Р. С. Громов | 16.03.2024 – 23.03.2024 |
| Испытания программы | Разработка, согласование и утверждение порядка и методики испытаний.  Корректировка программы и программной документации по результатам испытаний. | Р. С. Громов | 24.03.2024 – 30.03.2024 |
| 1. Внедрение | Подготовка и передача программы | Подготовка и передача программы и программной документации для сопровождения  Оформление и утверждение акта о передаче программы на сопровождение | Р. С. Громов | 31.03.2024 – 01.04.2024 |

Программа и документация к ней разрабатываются к утвержденным срокам сдачи курсового проекта (02.04.2024)

# **ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ**

## **Виды испытаний** Целью испытаний является выявления отклонений программы от функциональных требований технического задания. Испытания производятся согласно документу «Программа и методика испытаний (ГОСТ 19.301-78)», в котором указаны:

1. Порядок тестирования функциональных требований.
2. Способы и методики тестирования функциональных требований.
3. Порядок оценивания интерфейса на предмет соответствия техническому заданию.
4. Методика оформления результатов тестирования. Сроки проведения испытаний и ознакомления с их результатами оговариваются отдельно.
   1. **Общие требование к приёмке работы**

Контроль и приемка программного продукта осуществляются в соответствии с документом Программа и методика испытаний (ГОСТ 19.301-79). Прием программы будет утвержден при корректной работе программы в соответствии с пунктом 4.1.1. и при предоставлении полной документации к продукту, указанной в пункте 5.1, выполненной в соответствии с требованиями, указанными в пункте 5.2 данного технического задания.

**ПРИЛОЖЕНИЕ 1**

# **СПИСОК ИСПОЛЬЗУЕМОЙ 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 оформлениюc. // 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. Язык программирования Small-C

<https://en.wikipedia.org/wiki/Small-C>

1. Compilers: Principles, Techniques and Tools. Alfred V. Aho. <https://en.wikipedia.org/wiki/Compilers:_Principles,_Techniques,_and_Tools>
2. Конструирование компиляторов, учебное пособие. Сергей Свердлов. <https://pm.vogu35.ru/~c3c/CompilersConstruction/Sergey_Sverdlov_Compiler_Construction_(electronic%20version).pdf>

**ПРИЛОЖЕНИЕ 2**

# **ГЛОССАРИЙ**

1. Компилятор – это программа, переводящая написанный на языке программирования текст в набор машинных кодов
2. Компиляция – это процесс сборки программы, которая включает в себя трансляцию всех модулей программы, написанных на одном или нескольких исходных языках программирования высокого уровня и/или языке ассемблера, в эквивалентные программные модули на низкоуровневым языке, а также вставка в программу кода всех функций, импортируемых из статических библиотек
3. Препроцессинг – процесс начальной обработки текста исходной программы, включающая в себя удаление однострочных, многострочных комментариев, удаление пробелов, символов табуляции и перехода на новую строку
4. Лексический анализатор – часть компилятора, которая проходит по отформатированному исходному коду программы и разбивает её на последовательность токенов
5. Токен – абстрактная пара, состоящая из имени и атрибута. Представляет из себя тип лексический единиц для языка программирования
6. Парсер (синтаксический анализатор) – часть компилятора, обрабатывающая последовательность токенов, строящая абстрактное синтаксическое дерево и выявляющая синтаксические ошибки в тексте исходной программы, которые впоследствии передаются лексическому анализатору
7. Абстрактное синтаксическое дерево – это абстрактная структура данных, являющаяся конечным ориентированным деревом, в котором внутренние вершины сопоставлены с операторами языка, а листья – с соответствующим операндам
8. Семантический анализатор – часть компилятора, которая обходит абстрактное синтаксическое дерево для обнаружения в нём семантических ошибок (логических ошибок) и ошибок, происходящих во время выполнения программы

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

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