МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 42

ОТЧЁТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель		С.Ю. Гуков
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

Системы контроля версий (VCS)

по курсу:

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №	4329		Д.С. Шаповалова
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2025

ОГЛАВЛЕНИЕ

Цель работы

Изучить предназначение и различные способы организаций систем контроля версий (Version Control System, VCS) Git. Познакомиться с операциями над файлами в репозитории и с приемами командной работы над проектом.

Постановка задачи

Задание можно выполнять в любой платформе, основанной на git (GitHub, GitLab, GitFlic, BitBucket и др.).

Необходимо объединиться в команды по 2-4 человека. У каждого участника команды должен быть свой зарегистрированный аккаунт на выбранной платформе. Один из участников команды создает репозиторий и присоединяет к нему остальных участников. Необходимо придумать общий интерфейс программы (один из участников делает коммит набросков интерфейса в репозиторий, остальные обновляют у себя локальную копию репозитория). Далее каждый из участников обязательно в своей отдельной ветке выполняет свое задание по варианту (задания в команде должны различаться), периодически делая коммиты своих классов и изменений в коде в репозиторий и делая pull request в ветку dev, при этом обновляя (дополняя) свой локальный проект кодом этой ветки (с обновлениями коллег по команде). Ветки после слияния удалять не нужно. Также при pull request каждому необходимо сделать проверку (review) кода коллег – оставить несколько комментариев с советами и замечаниями по различным местам в коде. После того, как все участники команды сделают свое задание, ветка dev сливается в главную ветку master (main), и оформляется файл README.md с пояснениями о выполненных заданиях.

В итоге должен получиться проект с единым интерфейсом, выполняющий несколько различных задач (по количеству участников команды). У каждого участника команды на компьютере должен находиться полный общий локальный проект (содержащий свое реализованное задание и код коллег по команде). В качестве проверки задания преподаватель также будет смотреть в онлайн репозитории созданные ветки и список коммитов – кто из участников, когда и какие сделал изменения в проекте.

Проект обязательно должен иметь графический пользовательский интерфейс (User Interface, UI), а также может быть написан на любом языке программирования.

Замечание: разрешается выполнять проект в одиночку, при условии имитации работы в команде (регистрации 2-3 аккаунтов и выполнении различного задания с каждого аккаунта).

Требования к структуре проекта

- ☐ Применение принципов ООП (наследования, инкапсуляции, полиморфизма, абстракции) и SOLID
- ☐ Дружелюбный графический пользовательский интерфейс
- ☐ Реализация своего задания по варианту
- ☐ Наличие общего связанного пользовательского интерфейса на команду
- ☐ Наличие как минимум следующих веток: master (main), dev, собственной ветки каждого участника команды
- ☐ Разные коммиты (с осмысленным описанием изменений) от разных участников команды, от каждого минимум по 5 штук
- ☐ Наличие pull request с описанием изменений и комментариями коллег по команде
- ☐ Хорошей ситуацией будет, если будут возникать конфликты (можно вставить скриншоты в отчет)

Вариант 17. Пользователь открывает файл с данными о проценте детей, рожденных вне брака за последние 15 лет. Вывести эту информацию на экран в удобном табличном формате. По этим данным построить графики зависимости от года. Вычислить максимальный и минимальный процент изменения этих данных за год. Реализовать статистическое прогнозирование методом экстраполяции по скользящей средней на последующие N лет, вывести эту информацию на отдельном графике либо закрасить другим цветом на том же.

Краткое описание хода разработки и назначение используемых технологий

Ход разработки:

1. Сделан репозиторий на гитхаб, добавлены соавторы.
2. Разработана логика функции со статистикой рождения детей вне брака.
3. Разработан интерфейс для этой функции.
4. Разработан общий интерфейс программы.

Назначение используемых технологий

- python tkinter - для отображения интерфейса,
- python matplotlib - для отрисовки графиков.

Исходный код программы (с комментариями в необходимых местах)

Представлен на платформе гитхаб: urushicries/laba3: rep for 3rd lab of tech of programming

В ходе разработки были созданы ветки: main, dev, danya, dasha.

Ветка main отвечает за публичную версию проекта.

Ветка dev представляет разрабатываемую версию и отображает итог слияния личных веток разработчиков (danya и dasha).

Ветка danya принадлежит пользователю urushicries, студенту группы 4329 Онопричуку Данилу.

Ветка dasha принадлежит пользователю MyataEtoki, студентке группы 4329 Шаповаловой Дарье.

Скриншоты с сервиса git (ветки, коммиты, слияния и т.д.)

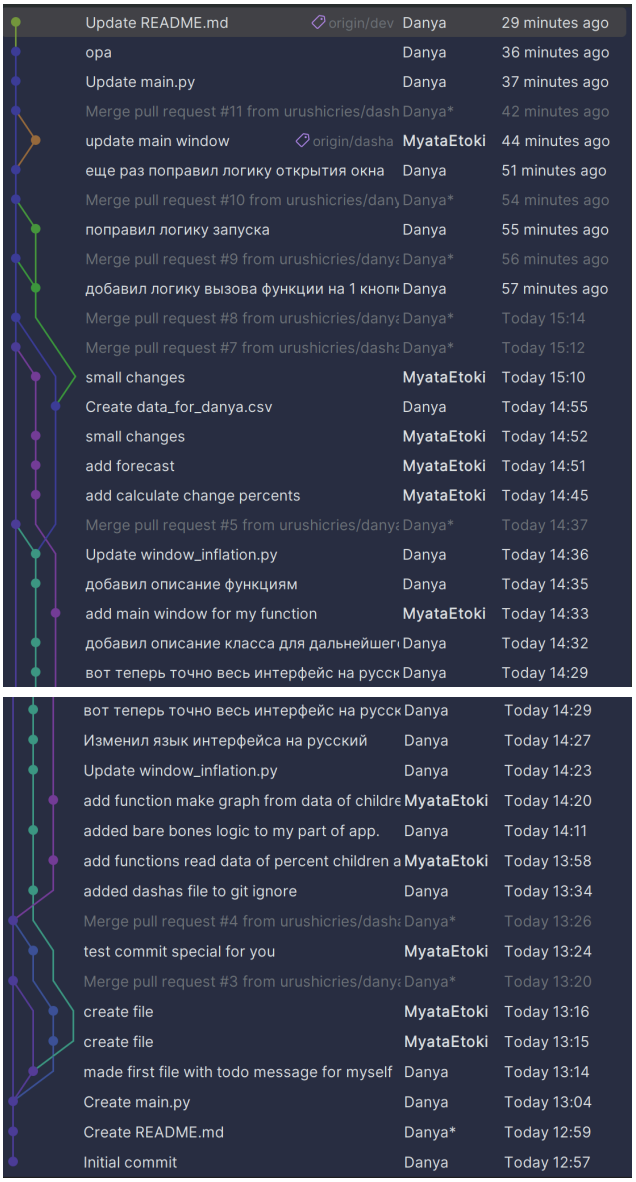

Рисунок 1.1 - Схема разработки, с отображением веток danya и dasha, коммитами, слияниями.

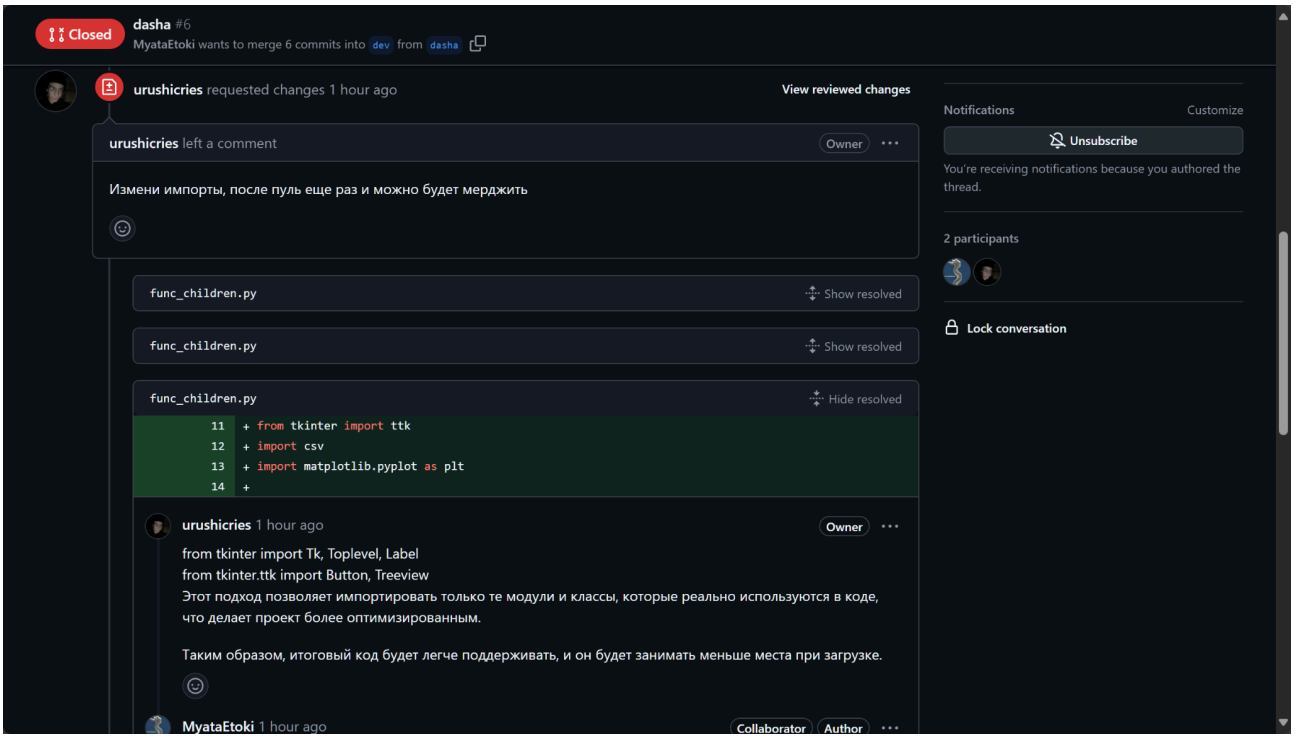

Рисунок 1.2 - Скриншот code review при pull request.

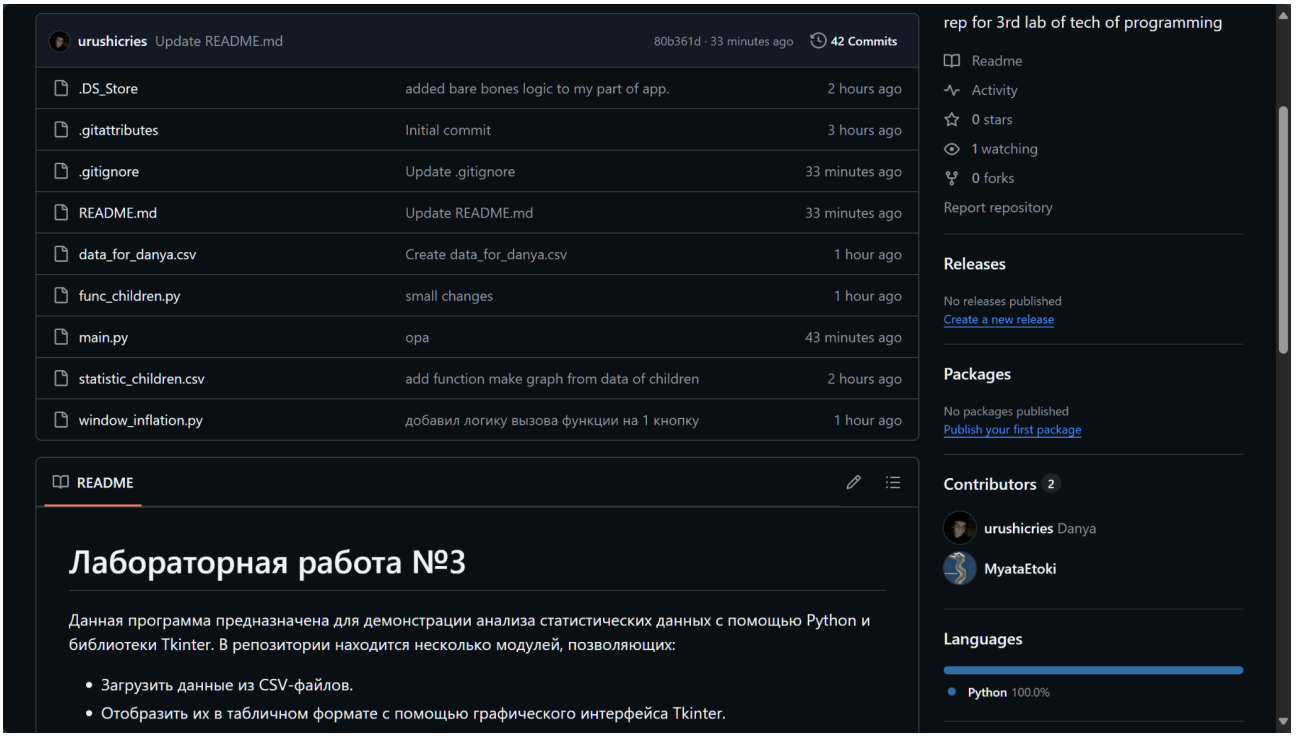

Рисунок 1.3 - Скриншот репозитория

Результаты работы программы с примерами разных сценариев (скриншоты)

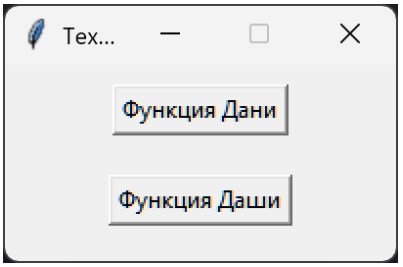

Рисунок 2.1 - Общий главный интерфейс

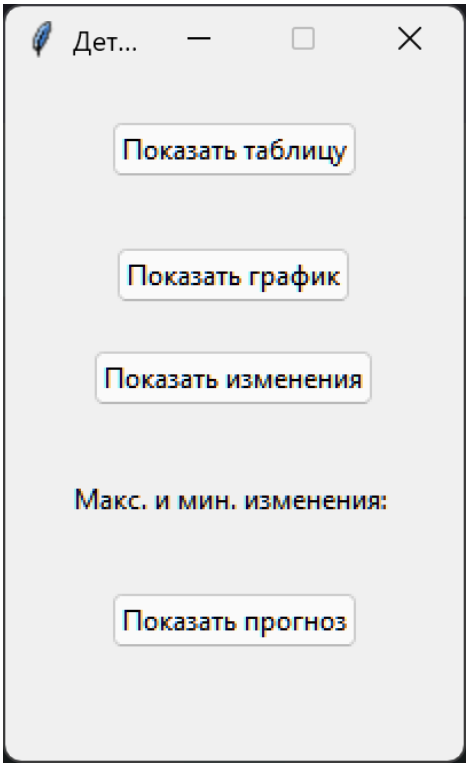

Рисунок 2.2 - Главный интерфейс моей функции

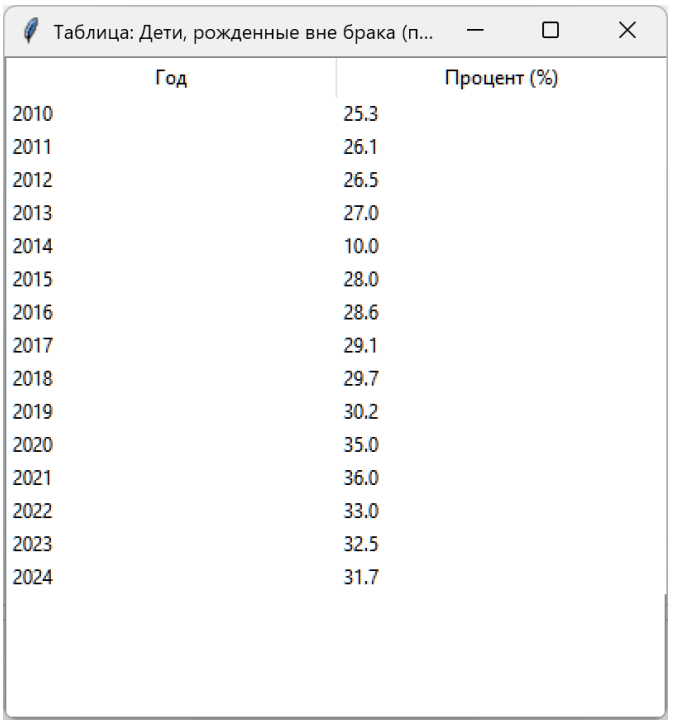

Рисунок 3.1 - Отображение таблицы

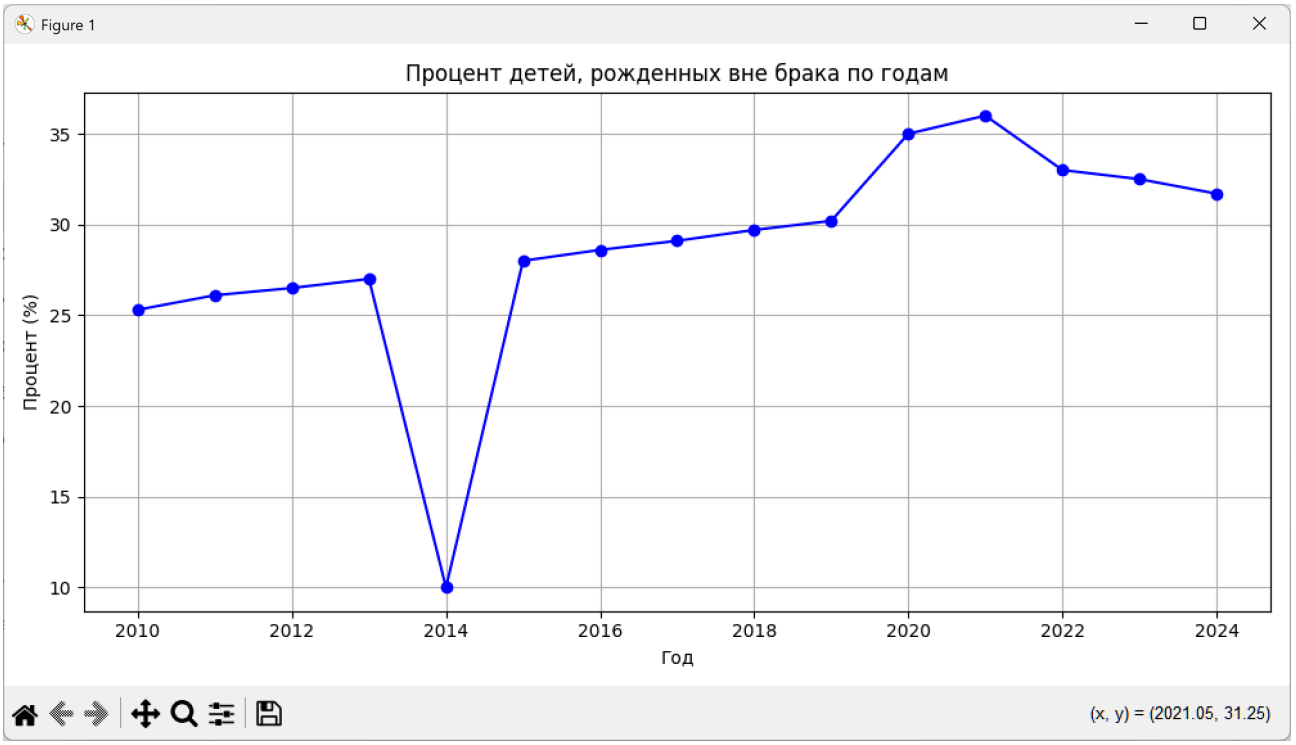

Рисунок 3.2 - Отображение графика

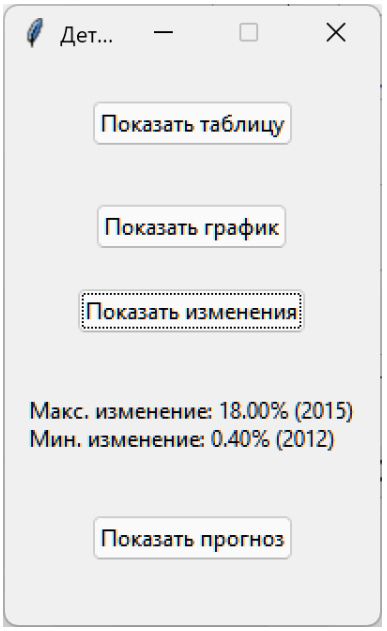

Рисунок 3.3 - Отображение максимального и минимального процента изменений

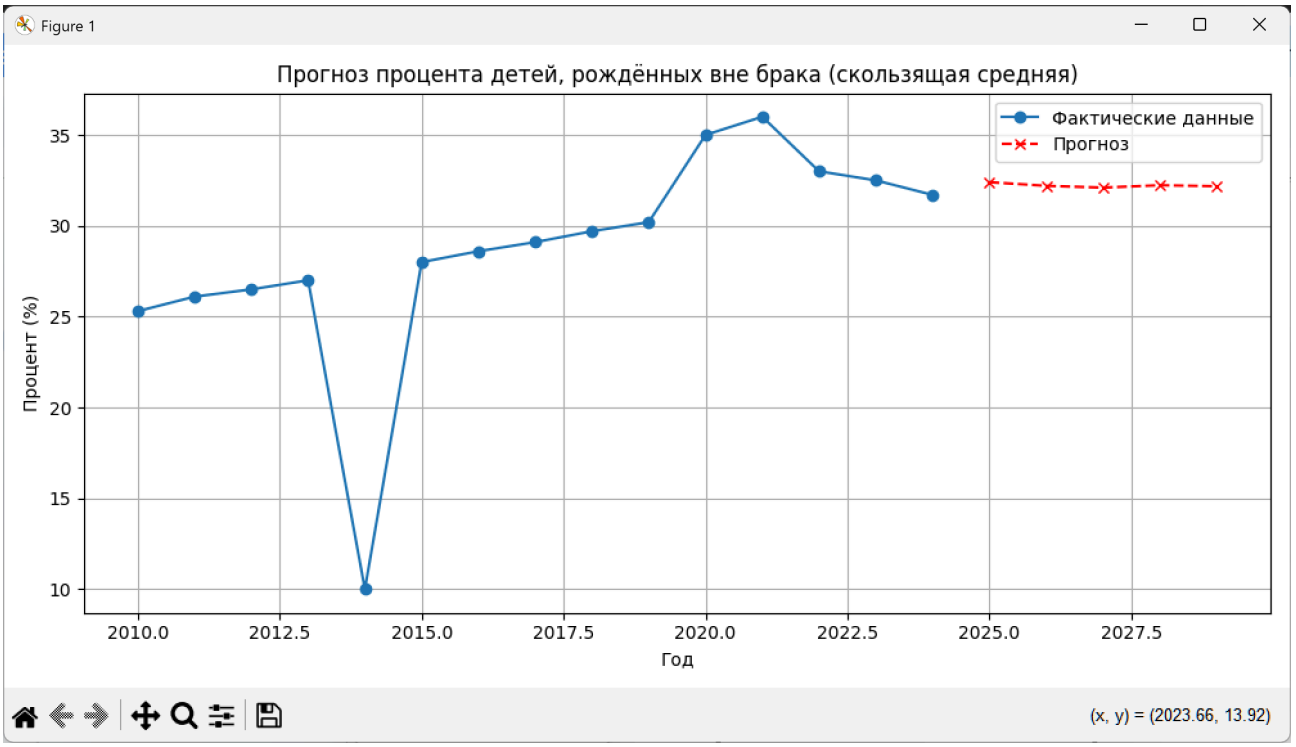

Рисунок 3.4 - Отображение графика прогноза

Выводы

В ходе выполнения лабораторной работы мной были освоены и изучены: основы работы с системой контроля версий git в команде; понятия commit, pull request, code review, brunch; библиотеки tkinter, matplotlib. Написанная программа была протестирована, полученный результат соответствует ожиданиям.