* **Волчий остров(Сивяков).** Волчий остров размером 20x20 заселен дикими кроликами, волками и волчицами. Имеется по нескольку представителей каждого вида. Кролики довольно глупы: в каждый момент времени они с одинаковой вероятностью 1/9 передвигаются в один из восьми соседних квадратов (за исключением участков, ограниченных береговой линией) или просто сидят неподвижно. Каждый кролик с вероятностью 0,2 превращается в двух кроликов. Каждая волчица передвигается случайным образом, пока в одном из соседних восьми квадратов не окажется кролик, за которым она охотится. Если волчица и кролик оказываются в одном квадрате, волчица съедает кролика и получает одно очко. В противном случае она теряет 0,1 очка. Волки и волчицы с нулевым количеством очков умирают. В начальный момент времени все волки и волчицы имеют 1 очко. Волк ведет себя подобно волчице до тех пор, пока в соседних квадратах не исчезнут все кролики; тогда, если волчица находится в одном из восьми близлежащих квадратов, волк гонится за ней. Если волк и волчица окажутся в одном квадрате и там нет кролика, которого нужно съесть, они производят потомство случайного пола. Запрограммировать предполагаемую экологическую модель и понаблюдать за изменением популяции в течение некоторого периода времени.
* **Задача об инфекции стригущего лишая.** Промоделировать процесс распространения инфекции — стригущего лишая по участку кожи размером *п × п (п* — нечетное) клеток. Предполагается, что исходной зараженной клеткой кожи является центральная. В каждый интервал времени пораженная инфекцией клетка может с вероятностью 0,5 заразить любую из соседних здоровых клеток. По прошествии шести единиц времени зараженная клетка становится невосприимчивой к инфекции, возникший иммунитет действует в течение последующих четырех единиц времени, а затем клетка оказывается здоровой. В ходе моделирования описанного процесса выдавать текущее состояние моделируемого участка кожи в каждом интервале времени, отмечая зараженные, невосприимчивые к инфекции и здоровые клетки.
* **Построение с помощью циркуля и линейки.()** Составить программу, автоматизирующую процесс построения фигур на плоскости с помощью циркуля и линейки. Программа должна уметь выполнять следующие команды:
* отметить произвольную точку и обозначить ее;
* построить прямую, проходящую через две точки;
* построить произвольную прямую;
* построить окружность с заданным центром данного радиуса;
* построить и обозначить точку пересечения двух линий.

Программа должна содержать 10—15 стандартных задач на построение школьного курса геометрии, предлагать их для решения и контролировать процесс построения и полученное решение.

* **«Морской бой»(Тигранян).** Составить программу для игры в морской бой игрока с компьютером. Программа должна позволять расставлять корабли на поле 10*×*10, контролировать правильность их расстановки, давать противникам возможность поочередно делать ходы и выдавать соответствующие информационные сообщения. Когда в качестве одного из игроков выступает компьютер, программа должна анализировать предыдущие ходы и следующий делать на основе проведенного анализа.
* **Обучающе-контролирующая программа «Сложение и вычитание отрицательных чисел».** Составить программу, обучающую учащихся 6 класса сложению и вычитанию отрицательных чисел, а также предлагающую серию заданий различной сложности для закрепления навыков действий над такими числами.
* **«Математико» (итальянская игра).** Имеется набор из 52 карточек, на которых записаны числа от 1 до 13, причем карточки с каждым из этих чисел встречаются четырежды. Разработать программу, которая позволяет имитировать игру человека с компьютером. Имеется квадратное поле с 25 клетками. Программа случайным образом извлекает какую-либо из имеющихся карточек и выдает записанное на ней число. Каждый игрок заносит это число в одну из клеток квадрата. Так продолжается до тех пор, пока не будут заполнены все клетки квадрата.



По окончании игры заполнение соответствующего квадрата оценивается определенным количеством очков. Цель игры — разместить числа в клетках так, чтобы набрать наибольшее количество очков в соответствии с данной таблицей



Разработать для компьютера оптимальную стратегию заполнения квадрата.

* **Карточные игры(Чирва).** Составить программу, которая раздает игральные карты заданному количеству игроков (одним из игроков является человек, за остальных играет компьютер) и моделирует игру в «дурака». Компьютерная программа играет случайным образом, без анализа уже вышедших карт. Количество игроков не превышает шести.
* **«Крестики-нолики»(Лицкевич).** Составить программу, позволяющую играть на бесконечном поле в «крестики-нолики»:

а) игроку с компьютером;

б) двум игрокам.

Если в качестве игрока выступает компьютер, программа делает первый ход. Делая очередной ход, программа анализирует ситуацию, рассчитывая возможные ходы противника вперед на 1 - 2 хода, и в результате проведенного анализа поступает оптимальным образом.

* **«Быки и коровы».** Составить программу, позволяющую играть в «Быки и коровы»:

а) игроку с компьютером;

б) двум игрокам.

Каждый из противников задумывает четырехзначное число, все цифры которого различны (первая цифра числа отлична от нуля). Необходимо разгадать задуманное число. Выигрывает тот, кто отгадает первый. Противники по очереди называют друг другу числа и сообщают о количестве «быков» и «коров» в названном числе («бык» — цифра есть в записи задуманного числа и стоит в той же позиции, что и в задуманном числе; «корова» — цифра есть в записи задуманного числа, но не стоит в той же позиции, что и в задуманном числе). Например, если задумано число 3275 и названо число 1234, получаем в названном числе одного «быка» и одну «корову». Очевидно, что число отгадано в том случае, если имеем 4 «быка».

* **«Числовые головоломки».** Составить программу, которая:

1) предлагает игроку числовые головоломки типа **ОДИН + ОДИН= МНОГО** из некоторого набора таких головоломок (до 30);

2) позволяет решить головоломку;

3) контролирует правильность решения.

* **«Графики».** Составить программу, которая предлагает пользователю некоторый список функций для построения графиков (например, *у* = *ах2 + bх + с, у = asin х + b* и т.д. — до 25 наименований). После выбора соответствующей функции, задания коэффициентов и отрезка, на котором выполняется построение, программа строит указанный график. Затем значение коэффициентов и положение графика можно менять (например, с помощью клавиш управления курсором), после чего график перестраивается и записывается обновленное уравнение соответствующей кривой.
* **Игра «Две лисы и 20 кур»(Кабанова).** На поле указанной формы



находятся две лисы и 20 кур. Куры могут перемещаться на один шаг вверх, влево или вправо, но не назад и не по диагонали. Лисы также могут перемещаться только на один шаг (вверх, вниз, влево и вправо). Лиса может съесть курицу, как в игре в шашки: если в горизонтальном или вертикальном направлении за курицей на один шаг следует свободное поле, то лиса перепрыгивает через курицу и берет ее. Лисы всегда обязаны есть, и, когда у них есть выбор, они обязаны осуществлять «наиболее длинное поедание». Если два приема пищи имеют одинаковую длину, осуществляется один из них — по выбору лисы. Составить программу, которая играет за лис. Игрок перемещает кур. Партнеры играют по очереди, причем куры начинают. Они выигрывают партию, если девяти из них удается занять 9 полей, образующих верхний квадрат поля. Начальное положение кур и лис изображено на рис. Лисы выигрывают, если им удается съесть 12 кур, так как тогда оставшихся кур недостаточно, чтобы занять 9 верхних полей.

* **Программа «Игры со спичками»(Гайдаржи).** Составить программу, которая:

1) предлагает игроку головоломки со спичками из некоторого набора таких головоломок (до 30 штук);

2) позволяет решить головоломку, передвигая спички;

3) контролирует правильность решения.

* **Игра в слова (Даниил Попов).** Составить программу, позволяющую компьютеру и человеку играть в слова. Предварительно программа объясняет правила игры и позволяет уточнить их в любой момент. Тематикой игры могут быть по выбору города, животные, растения и т.д. Тематику из предложенных компьютером (не менее 5). выбирает человек. Для игры компьютер использует собственную базу данных (для каждой тематики свою), хранящуюся в виде текстового файла. Если названное человеком слово отсутствует в базе, уточняется, правильно ли оно названо, и в случае правильности заносится в базу. Правила игры: первый игрок называет слово, а второй должен предложить другое, начинающееся с той буквы, на которую оканчивается названное.
* **Ребусы.** Выбрав какой-либо школьный предмет (информатика, математика и т.д.), подобрать ребусы по нему и предложить их для решения. Программа должна позволять выбрать тот или иной ребус, проконтролировать его решение и подвести итоги при завершении работы.
* **Инженерный калькулятор.** Два режима ввода: встроенная клавиатура или поле ввода строки (например, 5+sqrt(sin(3\*tg(4))+10^(45))). Основные требования к программе:

1) за основу взять калькулятор Windows 7 (все функции, 4 вида: инженерный, статистика, программист, обычный). При этом стараться не использовать стандартные функции (тригонометрические, возведение в степень и т.д.)

2) разбор строки с выражением, проверка на корректность, в случае ошибки, подробное описание ошибки и подсказка к устранению ошибки. Например, "Ошибка, неизвестная функция arcsin. Возможно, предполагалось arsin";

3) нахождение определенных интегралов (не менее пяти численных методов, включая метод Монте-Карло), решение нелинейных уравнений (не менее пяти методов);

4) операции с матрицами, нахождение определителей, решение системы линейных уравнений;

5) построение графиков функций, интерполяция, сплайн, прогнозирование.

6) разрешить тонкую настройку калькулятора с выбором методов вычисления;

7) применение ООП (классы, наследование, перегрузка операторов, шаблоны)

* **Графический редактор(Колесников)**. Требования к программе:

1) Неполный аналог графического редактора Paint в Windows 7.

2) Создание отдельного режима работы редактора для создание рисунков к геометрическим задачам (аналог программы "Живая Геометрия")

3) Нахождение численных характеристик фигур, нарисованной пользователем (например, площади).

* **Шифрование данных**. Основные требования к программе:

1) Шифрование и дешифрование данных с возможностью настройки методов шифрования. Общее количество алгоритмов шифрования не менее пяти. Для реализации алгоритмов использовать собственный класс для работы с длинными числами (причем предложить два варианта такого класса в зависимости от выбора способа хранения цифр длинного числа).

2) Стеганографические алгоритмы. Скрытие данных в текстовом файле, скрытие данных в изображениях, скрытие данных в аудиофайлах.

3) Атаки на криптосистемы (в частичной или полной неизвестности).

* **Распознавание образов**. Основные требования к программе:

1) Распознавание графических примитивов на изображениях (круг, квадрат, треугольник и др.)

2) Распознавание одиночных символов (буквы, цифры и др.), нарисованные пользователем.

3) Распознавание слов, состоящих из букв и цифр известного алфавита. Для тестирования алгоритмов создать класс для генерации таких слов на картинках с возможностью настройки параметров: размер букв, длина слова, шрифт, цвет, искажения и др.

4) Протестировать алгоритмы из предыдущих пунктов на реальных изображениях. Модифицировать алгоритмы для конкретных данных.

5) Распознавание лиц на фотографиях. Определение областей, в которых наиболее вероятно изображены лица. Определение данных человека по фотографии (сравнение с базой данных фото претендентов)

6) Дополнить программу некоторыми функциями программ распознавания текстов (например, FineReader).

7) По данным изображениям территорий с деревнями (вид сверху) определить количество домов.

* **Анализ алгоритмов сортировки данных**. Требования к программе:  
  1) Реализация алгоритмов сортировки (не менее двадцати). Создание собственных классов для всех структур, применяемых в алгоритмах. Тестирование алгоритмов на различных входных данных. Построение графиков зависимостей времени от типов данных, от объема данных, от вида данных.

2) Предельные значения объема входных данных при фиксированной верхней границе времени сортировки. Многопоточное программирование.

3) Получение алгоритмы выбора наиболее подходящего алгоритма сортировки для конкретных данных.

* **Работа с текстом**. Основные требования к программе:

1) Создать текстовый редактор с базовыми функциями: работа с несколькими файлами, автоматическое сохранение черновиков, поиск с настройками (замена, регистр, неточное совпадение), настройка представления текста, изменение кодировки и др.

2) Поиск по тексту реализовать по нескольким алгоритмам, создав для них собственные классы.

3) Возможность определять стиль данного текста (научный, публицистический и др.). Разделение данного текста на фрагменты различного стиля.

4) Определение автора данного текста, если дана база текстов нескольких авторов.

5) Определение уникальности (в %) данного текста в сравнении с базой текстов. Понятие уникальности определить самостоятельно (несколько вариантов). Выявить наиболее эффективное определение для данной задачи.

6) Проверка данного текста на орфографические и пунктуационные ошибки с возможностью автоматического исправления.

7) Поиск одинаковых слов в одном предложении или близких предложениях с заменой на синонимы. Возможность автоматической замены числа из цифр на число из слов.  
8) Подбор рифмы для данного фрагмента стихотворения с пропущенным словом.

* **Моделирование биологических систем.** Основные требования к программе:  
  1) Реализация игры "Жизнь". Классификация различных начальных состояний игры.  
  2) Моделирование взаимодействия популяций с различными характеристиками. Приведем пример для двух популяций: на плоскости расположены две группы точек (красные и синие). Каждая точка характеризуется несколькими числовыми показателями (возраст, сила, здоровье, смелость, скорость и др.). Группы движутся по определенным правилам, например, одна (травоядные) в сторону некоторой области на карте (место нахождение пищи). Другая (хищники) в сторону центра тяжести группы травоядных. В случае, когда две точки из разных групп сходятся достаточно близко (менее данного числа), происходит подсчет количества точке вокруг них в окружности данного радиуса. Если количество хищников превосходит количества травоядных, то точка либо отдаляется от хищников, либо происходит схватка, результат которой зависит от здоровья, смелости, возраста, силы противников и случайных факторов. Программа моделирует данное взаимодействие с целью определения различных зависимостей. Например, соотношения размеров популяции и суммарного показателя силы популяции и исхода взаимодействия.

3) Моделирование взаимодействия произвольного количества популяций. Отношения между популяциями: вражеские, дружеские, нейтральные.  
4) Моделирование популяций, в каждой из которых есть лидер, от поведения которого зависит состояние популяции. Введите в рассмотрение области на карте, нахождение в которых позволяет популяции аккумулировать здоровье и силу.  
5) Моделирование взаимодействия популяций по дифференциальным уравнения (модель Лотки-Вольтерра и др.).

* **Чат(Иванов Ян)**. Основные требования к программе:

1) + Работа в локальной сети.

2) + Регистрация пользователей (администратор, модератор, пользователь). Возможность загрузки фото пользователя - добавить загрузку

3) + Вывод списка онлайн-пользователей, возможность просмотра учетных данных пользователя - добавить страницу просмотра

4) + История сообщений пользователя (хранится на клиенте). История сообщений с модератором хранится и на сервере.

5) + Возможность блокировки пользователя модераторов на указанное время. - добавить выбор времени

6) + Настройка представления текста (шрифты, цвет и т.д.). - поправить размер вынести на отдельную страницу

7) + Возможность автоматического транслита (например, "привет" в "privet").

8) - Возможность обмена файлами между пользователем.

9) - Шифрование данных.

10) + Приватная беседа, создание комнаты для нескольких пользователей.

11) + Автоматическая замена запрещенных слов на \*\*\*\*\*\*\*\*.

12) + - Автоматическая блокировка за спам (количество сообщений в единицу времени превысило лимит).

* **Танки(Матросов)**. Основные требования к программе:

1) Игра по сети или локально. Если локально, то с возможностью игры двух человек в команде с танками компьютера против другой команды из нескольких танков компьютера.

2) На карте периодически появляются бонусы (здоровье, снаряды, скорость и др.). Периодичность и место появление (например, чаще около противника) зависит от сложности раунда.

3) Доступна возможность создавать собственную карту для игры.

4) Танк компьютера может играть в нескольких стилях. Например, "боязливый" старается стрелять издали и собирать побольше бонусов, "контуженный" начинает разбивать стены или выдвигается в атаку при любом количестве танков противника впереди.

5) Кроме прохождения игры по сценариям доступен режим Survival.

6) В процессе игры танк зарабатывает очки, которые можно потратить на покупку новой модели танка или улучшение показателей модели текущей версии.

7) Возможность сохранения/загрузки игры.

8) Возможность запуска игры без участия танка человека и анализ процесса игры с точки зрения биологических популяций. Сбор и анализ статистики должен привести к решению задачи прогнозирования исхода игры по начальным данным.

* **Морской порт.** Требуется создать компьютерную модель обслуживания потока заявок на разгрузку, поступающих от грузовых судов (сухогрузов и танкеров), прибывающих в морской порт. Грузовые суда прибывают в порт согласно расписанию, но возможны опоздания и досрочные прибытия. Расписание включает день и время прибытия, название судна, вид груза и его вес, а также планируемый срок стоянки в порту для разгрузки.

Для разгрузки судов в порту используются три вида разгрузочных кранов, соответствующих трем видам грузов: сыпучим и жидким грузам, контейнерам. Число разгрузочных кранов каждого вида ограничено, так что поступающие заявки на разгрузку одного вида груза образуют очередь. Длительность разгрузки судна зависит от вида и веса его груза, а также некоторых других факторов, например, погодных условий. Любой дополнительный (сверх запланированного срока) день стояния судна в порту (из-за ожидания разгрузки в очереди или из-за задержки самой разгрузки) влечет за собой выплату штрафа (например, 2 тыс. у.е. за каждый дополнительный день простоя судна).

При моделировании прибытия судов отклонение их от расписания рассматривается как случайная величина с равномерным распределением в некотором интервале (например, от -2 до 9 дней). Еще одной случайной величиной, изменяющейся в фиксированном диапазоне (например, от 0 до 12 дней), является время задержки окончания разгрузки судна по сравнению с обычным (зависящим только от вида груза и его веса).  
Цель моделирования работы морского порта – определение для заданного расписания прибытия судов минимально достаточного числа кранов в порту, позволяющего уменьшить штрафные суммы. Период моделирования – месяц, шаг моделирования – 1-3 дня. В параметры моделирования следует включить расписание прибытия судов, количество кранов каждого вида, диапазоны разброса случайных величин (отклонения от расписания прибытия и отклонения от обычного времени разгрузки), а также шаг моделирования.  
Визуализация моделируемого процесса должна предусматривать показ очередей у разгрузочных кранов, приход судов в порт и их отход после разгрузки. Должен быть показан также список произведенных разгрузок, в котором указывается название разгруженного судна, время его прихода в порт и время ожидания в очереди на разгрузку, время начала разгрузки и ее продолжительность. По окончании моделирования должна быть выведена итоговая статистика: число разгруженных судов, средняя длина очереди на разгрузку, среднее время ожидания в очереди, максимальная и средняя задержка разгрузки, общая сумма выплаченного штрафа.