

Министерство образования Красноярского Края
Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей IT-Куб г. Норильск»

СОГЛАСОВАНО:

Председатель методического совета

 Грицюк Н. В.

протокол № 12

от « 3 » июня 2024 г.

УТВЕРЖДАЮ:

Директор филиала

 Дыптан Е.А.

Приказ 02-22-59

от « 9 » июня 2024 г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА НА 2024 – 2025 уч. г**
технической направленности

«Олимпиадное программирование на языке C++»

Срок реализации: 1 год
Возраст детей: 14-17 лет
Составитель программы:
Коньшин И.В., педагог
дополнительного образования

г. Норильск, 2024 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная программа «Олимпиадное программирование на C++» (далее - программа) имеет техническую направленность, и разработана в соответствии с основными нормативно-правовыми документами: Федеральным Законом «Об образовании» от 29.12.2012 г. № 273-ФЗ; Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам от 09.11.2018 г. № 196; Целевой моделью развития региональных систем дополнительного образования детей от 03.09.2019 г. № 467; Концепцией развития дополнительного образования детей до 2030 года утвержденной распоряжением Правительства Российской Федерации от 31.03.2022 №678-р; Санитарно-эпидемиологическими требованиями к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи от 28.09.2020 г. № 28.

Уровень программы — базовый, что предполагает углубление интересов, расширение спектра специальных знаний в области программирования, обеспечивает трансляцию общей и целостной картины в рамках содержательно-тематического направления программы.

Назначение программы — привлечь школьников к олимпиадной, интеллектуальной и творческой деятельности. В рамках курса обучающиеся овладеют продвинутыми навыками работы с языком C++, научатся эффективно применять различные структуры данных и алгоритмы, а также разовьют креативное мышление при решении сложных задач.

Курс нацелен на повышение результативности участников обучения в олимпиадах по программированию. Для этого были выбраны темы, которые встречаются в каждом или почти каждом контексте по олимпиадному программированию. Программа рассчитана на людей, которые уже умеют программировать и собираются участвовать в этапах Всероссийской олимпиады школьников и иных олимпиадах по программированию. В рамках реализации настоящей программы будут даваться не только учебные задачи, но и задачи с настоящих соревнований по программированию.

1.1. НОВИЗНА ДООП

Программой предусмотрены новые методики преподавания, в том числе – гибридное обучение и нововведения в формах диагностики, подведении итогов реализации программы, выполняемые в формате Всероссийских и Международных олимпиад по программированию.

В настоящее время Всероссийская олимпиада школьников по информатике проводится как олимпиада по программированию, принимать в ней участие могут учащиеся 8 – 11 классов, но программирование в рамках курса информатики изучается только в 10-11 классах, в лучшем случае в 9-10 классе. Основная масса участников Всероссийской олимпиады школьников по информатике это учащиеся 9-11 классов, учащиеся 8 классов участие в олимпиаде не принимают, потому что не знают ни одного языка программирования. Решить данную проблему позволяет предложенная программа.

1.2. АКТУАЛЬНОСТЬ ДООП

Происходящая цифровизация экономики и всего современного общества как никогда показала потребность практически всех отраслей человеческой деятельности в квалифицированных кадрах, владеющих навыками и умениями программирования. Проводящиеся как внутри страны, так и за ее пределами турниры и олимпиады по программированию подталкивают детей к постоянному развитию и совершенствованию своих знаний, что и является несомненным преимуществом этой дополнительной общеобразовательной общеразвивающей программы. Программа уникальна тем, что позволяет за короткий срок ее непрерывной интенсивной реализации погрузить ребенка в мир современного продвинутого программирования, его алгоритмов, математических и интеллектуальных моделей.

В конечном счете, занятия по программе «Олимпиадное программирование на языке C++» направлены на отбор талантливых учащихся и их интеграцию в интеллектуальный резерв государства.

1.3. ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ

Важной и педагогически целесообразной чертой программы являются регулярные олимпиадные симуляции, где участники могут применить полученные знания на практике и оценить свой прогресс. Так же программа включает в себя развитие навыков программирования на C++, с упором на темы, необходимые для решения олимпиадных задач. Курс создан для тех, кто стремится достичь выдающихся результатов в олимпиадах по программированию и раскрыть свой потенциал в этой увлекательной области.

Задача педагога – стимулировать учащихся на приобретение навыков, необходимых для самостоятельной практической и теоретической работы, а также для последующего успешного решения олимпиад по программированию. В этом контексте подчеркивается важность формирования у обучающихся ключевых навыков, способствующих их полноценному вовлечению в образовательный процесс.

При составлении программы учитывались следующие психофизиологические особенности потенциальных обучающихся:

- потребность в жизненном самоопределении и обращенность планов в будущее, осмысление с этих позиций настоящего;
- становление социальных мотивов гражданского долга;
- тенденция к осознанию школьником своего мировоззрения;
- потребность в осознании себя как целостной личности;
- оценке своих возможностей в выборе профессии, в осознании своей жизненной позиции;
- интерес ко всем формам самообразования;
- избирательность познавательных мотивов, диктуемая выбором профессии;
- устойчивость интересов, их относительная независимость от мнения

окружающих.

Данная образовательная программа помогает в решении следующих актуальных педагогических задач:

- стимулирование инженерного мышления обучающихся;
- формирование необходимых навыков для дальнейшего использования и совершенствования знаний языка C++;
- стимулирование интереса обучающихся к участию в олимпиадах по программированию различного уровня сложности.

Применение структурированного и последовательного метода обучения с использованием большого количества задач различной сложности по всем пройденным темам дает возможность учащемуся самостоятельно мыслить, обдумывать используемые алгоритмы, добывать недостающую информацию и решать задачи. Реализация данной программы повышает мотивацию обучающихся, стимулирует на совершенствование собственных навыков, а тем самым и усовершенствованию личностных качеств.

1.4. ЦЕЛЬ ДООП

Целью реализации настоящей дополнительной общеразвивающей программы является совершенствование обучающимися теоретических знаний и практических навыков в области решения задач повышенной сложности и олимпиадных задач по программированию, достаточных для прохождения на региональный этап Всероссийской олимпиады школьников.

Задачи программы:

- закрепить и отработать важнейшие понятия алгоритмизации и программирования;
- отработать решение задач повышенной сложности и олимпиадных задач по программированию с использованием полученных знаний;
- сформировать систему специальных знаний, умений и навыков в области олимпиадного программирования;
- изучить динамические структуры данных;
- научиться управлять затрачиваемой памятью компьютера;
- познакомиться с олимпиадной математикой;
- освоить информационную культуру: ответственное отношение к информации с учетом правовых и этических аспектов ее распространения;
- сформировать потребность в самостоятельном приобретении и применении знаний из дополнительных источников;
- сформировать способность выдвигать и доказывать гипотезы опытным путем, разрабатывать стратегию решения, прогнозировать результаты своей деятельности, анализировать и находить рациональные способы решения задачи путем оптимизации, детализации созданного алгоритма;
- поощрять позитивное отношение к любым специализациям IT-отрасли и нацеливать на дальнейшее развитие в этой области

безотносительно к успехам или неудачам в области олимпиадного программирования.

1.5. ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП

Программа «Олимпиадное программирование на языке C++» рассчитана на обучающихся 14-17 лет. Максимальное количество обучающихся в группе – 12 человек.

1.6. УСЛОВИЯ ВХОЖДЕНИЯ В ДООП

Набор на программу осуществляется в соответствии с правилами приема и отчисления обучающихся АНО ДТ «Красноярский «Кванториум» в г. Норильске.

Обучающиеся, поступающие на программу, проходят входное тестирование и собеседование, направленное на выявление их индивидуальности и уровня владения основами программирования.

1.7. СРОК РЕАЛИЗАЦИИ ДООП

Программа рассчитана на 1 учебный год. Нагрузка на обучающегося составляет 144 часа за весь период обучения.

1.8. РЕЖИМ ЗАНЯТИЙ, ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ

Учебные занятия проходят в очной форме. Режим занятий – 2 раза в неделю по 2 академических часа (1 академический час - 40 минут) с обязательным перерывом.

При проведении занятий используются как стандартные формы работы, так и нетрадиционные:

- лекции: изложение преподавателем информации, необходимой для введения в новые темы и предоставления основных концепций;
- семинары: краткий повтор теоретической части лекции и ее практические групповые обсуждения, совместное решение проблемных задач;
- практические занятия: работа с конкретными задачами, необходимая для выработки практических умений;
- дискуссии: постановка спорных вопросов, отработка умения отстаивать и аргументировать свою точку зрения;
- круглый стол: неформальное обсуждение выбранной тематики;
- симуляция олимпиады;
- мозговая атака: решение нестандартных задач в коллективе.

Обучение проводится в комбинированном формате с проведением лекций и демонстраций для наилучшего усвоения теоретического материала. После лекционных занятий проводятся практические, направленные на закрепление полученных теоретических знаний.

1.9. ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ И

ЭФФЕКТЫ, СПОСОБЫ ПРЕДЪЯВЛЕНИЯ И ОТСЛЕЖИВАНИЯ РЕЗУЛЬТАТОВ

Предметные результаты:

- умение объяснять и использовать на практике конструкции любой сложности;
- уверенное владение синтаксисом C++;
- умение свободно применять стандартные библиотеки языка C++;
- обширный практический опыт программирования на языке C++;
- сформированный навык решения олимпиадных задач;
- умение эффективно использовать динамические структуры данных, такие как списки, деревья и графы, для решения различных задач;
- понимание принципов динамического программирования;
- освоение базовых алгоритмов олимпиадного уровня.

Метапредметные результаты:

- сформировано ответственное отношение к информации с учетом правовых и этических аспектов ее распространения.

Личностные результаты:

- сформирована потребность в самостоятельном приобретении и применении знаний из дополнительных источников;
- сформирована способность выдвигать и доказывать гипотезы опытным путем, разрабатывать стратегию решения, прогнозировать результаты своей деятельности, анализировать и находить рациональные способы решения задачи путем оптимизации, детализации созданного алгоритма.

Текущий контроль освоения программы проводится во время занятий при помощи наблюдений, опросов, выполнения практических работ. Для итоговой проверки полученных знаний используются публичные защиты результатов, полученных на практике, а также участие в научно-технических мероприятиях различного уровня.

В конце первого полугодия проводится промежуточная аттестация (2 часа) в виде симуляции олимпиады. Программой предусмотрено участие в реальных олимпиадах разного уровня.

Аттестация по итогам освоения программы также проходит в виде симуляции олимпиады. Для оценки финальных результатов работы рассматривается количество правильно решенных задач.

Для успешной аттестации по итогам освоения программы необходимо решить не менее половины задач.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика	Форма контроля
Раздел 1. Основы программирования и языка C++ (40 часов)					
1.1	Введение в программирование	2	2	0	
1.2	Ввод и вывод данных: работа с вводом-выводом, потоки данных.	4	2	2	
1.3	Основы C++: синтаксис, переменные, операторы, условные конструкции, циклы.	10	4	6	
1.4	Функции: объявление, параметры, возврат значений.	10	2	8	
1.5	Массивы и векторы: объявление, индексация, итерирование, многомерные массивы.	10	4	6	
1.6	Текущий контроль по разделу «Основы программирования и языка C++»	4	2	2	Самостоятельное решение олимпиадных задач начального уровня
Раздел 2. Алгоритмы и структуры данных (56 часов)					
2.1	Сортировки: пузырьковая, выбором, вставками, быстрая сортировка.	12	4	8	
2.2	Поиск: бинарный поиск, линейный поиск.	10	4	6	
2.3	Стеки и очереди: реализация, применение.	10	2	8	
2.4	Рекурсия: базовые понятия, рекурсивные алгоритмы.	10	4	6	
2.5	Структуры данных: списки, связанные списки, деревья, хеш-таблицы.	10	4	6	
2.6	Текущий контроль по разделу «Алгоритмы и структуры данных»	4	2	2	Самостоятельное решение олимпиадных задач среднего уровня
Раздел 3. Олимпиадная математика (44 часа)					
3.1	Геометрические задачи: работа с координатами, расстояния, площади, периметры.	10	6	4	
3.2	Динамическое программирование: базовые принципы, рекурсивное решение, оптимизация.	10	4	6	

3.3	Графы: обходы, кратчайшие пути, поиск в ширину и глубину.	10	6	4	
3.4	Префиксные суммы и другие предподсчеты: оптимизация запросов, решение задач на массивах.	10	4	6	
3.5	Текущий контроль по разделу «Олимпиадная математика»	4	2	2	Самостоятельное решение олимпиадных задач продвинутого уровня
4	Итоговая олимпиада	4	0	4	
	Итого	144	58	86	

3. СОДЕРЖАНИЕ ПРОГРАММЫ

Раздел 1. Основы программирования и языка C++ (40 часов)

Тема 1.1. Введение в программирование на C++ (2 часа)

Теория: правила техники безопасности, об особенностях соревнований по олимпиадному программированию, вердикты проверяющей системы, структура программы на языке C++, основные типы данных и операции с ними, возможности целочисленной арифметики.

Практика: знакомство со средой программирования.

Тема 1.2. Ввод и вывод данных: работа с вводом-выводом, потоки данных (4 часа)

Теория: языковые конструкции C++: cin, cout, потоки данных, операторы << и >>, чтение данных из файла, метод get(), инструменты для работы со строковыми потоками.

Практика: ввод и вывод данных для взаимодействия с пользователем, а также обработки входных и выходных данных задач, составление линейных алгоритмов, работа с файлами.

Тема 1.3. Основы C++: синтаксис, переменные, операторы, условные конструкции, циклы (10 часов).

Теория: языковые конструкции C++: синтаксис, переменные, операторы, условные конструкции, циклы, эффективный способ проверки простоты числа, разложение числа на простые множители, наибольший общий делитель (НОД), наименьшее общее кратное (НОК), однопроходные алгоритмы, оценка сложности алгоритма по времени исполнения.

Практика: использование эффективных алгоритмов проверки простоты числа и разложения на простые сомножители, использование НОД и НОК при решении задач, повышение эффективности циклических алгоритмов путем полного ухода от цикла (сворачивание по формуле), либо избавления от одного из вложенных циклов

Тема 1.4. Функции: объявление, параметры, возврат значений (10 часов)

Теория: объявление функций, передача параметров, возвращаемые значения, функции, определяемые пользователем, рекурсивные подпрограммы, функции и процедуры обработки строк.

Практика: структурирование кода с помощью функций для решения сложных задач.

Тема 1.5. Массивы и векторы: объявление, индексация, итерирование, многомерные массивы (10 часов).

Теория: работа с массивами, векторами, индексация, итерирование, динамические массивы (vector) в C++, префиксные суммы, сортировка подсчетом, стандартная сортировка в C++ (из библиотеки algorithm), в том числе с использованием компараторов, вывод массива на экран, задание массива

типизированной константой, перестановка элементов массива, поиск максимального и минимального элементов, сортировка массива.

Практика: сортировка массивов, использование префиксных сумм при решении задач, использование идей сортировки подсчетом при решении задач.

Текущий контроль по разделу «Основы программирования и языка C++»: теоретическое тестирование и самостоятельное решение олимпиадных задач начального уровня (4 часа).

Раздел 2. Алгоритмы и структуры данных (56 часов)

Тема 2.1. Сортировки: пузырьковая, выбором, вставками, быстрая сортировка (12 часов).

Теория: различные алгоритмы сортировки для эффективной обработки и сортировки данных.

Практика: реализация алгоритмов сортировки.

Тема 2.2. Поиск: бинарный поиск, линейный поиск (10 часов).

Теория: особенности алгоритмов бинарного и линейного поиска.

Практика: реализация алгоритмов поиска.

Тема 2.3. Стеки и очереди: реализация, применение (10 часов).

Теория: очередь и стек как частные случаи однонаправленного списка, доступ к вершине стека, удаление элемента из вершины, добавление элемента в вершину, бинарные деревья, обход дерева.

Практика: решение задач, требующих особой организации данных, задачи на обработку строк или обход деревьев.

Тема 2.4. Рекурсия: базовые понятия, рекурсивные алгоритмы (10 часов).

Теория: рекурсивные функции, базовые понятия рекурсии, глубина рекурсии и общее количество рекурсивных вызовов, быстрое возведение в степень, рисование самоподобных ломаных, элегантные и эффективные алгоритмы для решения сложных задач.

Практика: решение задач на рекурсию.

Тема 2.5. Структуры данных: списки, связанные списки, деревья, хеш-таблицы (10 часов).

Теория: динамические структуры данных, современная организация адресации оперативной памяти, особенности работы со списками, связанными списками, деревьями и хеш-таблицами.

Практика: реализация различных структур данных — списки, связанные списки, деревья, хеш-таблицы.

Текущий контроль по разделу «Алгоритмы и структуры данных»: теоретическое тестирование и самостоятельное решение олимпиадных задач

среднего уровня (4 часа).

Раздел 3. Олимпиадная математика (44 часа)

Тема 3.1. Геометрические задачи: работа с координатами, расстояния, площади, периметры (10 часов).

Теория: базовые процедуры, прямая линия и отрезок прямой, математические операции, работа с формулами расстояний, площадей и периметров, треугольник, многоугольник, выпуклая оболочка.

Практика: решение задач с использованием алгоритмов вычислительной геометрии и задач о прямоугольниках.

Тема 3.2. Динамическое программирование: базовые принципы, рекурсивное решение, оптимизация (10 часов).

Теория: разделение задачи на подзадачи и сохранение результата для последующего использования, задачи оптимизации, стратегии для решения сложных задач, нисходящий подход, подход «снизу вверх», оптимизация алгоритмов.

Практика: работа с ключевыми элементами динамического программирования — рекурсией и запоминанием.

Тема 3.3. Графы: обходы, кратчайшие пути, поиск в ширину и глубину (10 часов).

Теория: представление графа в памяти компьютера, поиск в графе в глубину и в ширину, деревья, стягивающие деревья, порождение всех каркасов графов, каркас минимального веса, метод Краскала и Прима, связность, циклы, кратчайшие пути, алгоритм Дейкстры, алгоритм Флойда, независимые и доминирующие множества, раскраски, методы приближенного решения задачи коммивояжера, метод локальной оптимизации, алгоритм Эйлера, алгоритм Кристофидеса.

Практика: решение задач на обработку графов.

Тема 3.4. Префиксные суммы и другие предподсчеты: оптимизация запросов, решение задач на массивах (10 часов).

Теория: массив префиксных сумм, индекс равновесия массива, оптимизация запросов.

Практика: работа с массивами, обработка больших объемов данных с помощью предподсчетов.

Текущий контроль по разделу «Олимпиадная математика»: теоретическое тестирование и самостоятельное решение олимпиадных задач начального уровня (4 часа).

Раздел 4. Итоговая олимпиада (4 часа).

Практическая работа: симуляция олимпиады, задания которой будут включать в себя все изученные ранее модули, а также методы оптимизации кода, необходимые для решения олимпиадных задач.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Материально – техническое обеспечение:

1. Стол преподавателя
2. Стул преподавателя
3. Стол обучающегося
4. Стул обучающегося
5. Рабочая станция преподавателя
6. Ноутбук обучающегося
7. Интерактивная доска
8. Проектор
9. МФУ
10. Точки подключения к электрической сети

Программное обеспечение:

1. Операционная система Windows 10
2. Пакет программ MS OFFICE
3. Visual Studio C++.

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

- 1) Томас Х. Кормен, Алгоритмы. Вводный курс. – Вильямс. 2014. – 208 с.
- 2) Стефан Рэнди Дэвис С++ For Dummies, . – Диалектика. 2018. – 336 с.
- 3) Липпман С.Б, Лажоие Ж, Му Б.Э. Язык программирования С++. Базовый курс. – Вильямс. 2014. – 1120 с.
- 4) Густокашин М. Курс лекций по олимпиадной информатике. [Электронный ресурс] URL: <https://informatics.msk.ru/mod/resource/view.php?id=1381> (Дата обращения: 21.03.2022)
- 5) А. Лааксонен, Олимпиадное программирование. Изучение и улучшение алгоритмов на соревнованиях. – ДМК Пресс. 2020.
- 6) Великович Л.С., Цветкова М.С. Программирование для начинающих. – М. Бином. Лаборатория знаний, 2007. – 287 с.
- 7) Волченков С.Г., Корнилов П.А., Белов Ю.А. и др. Ярославские олимпиады по информатике. Сборник задач с решениями. – М. Бином. Лаборатория знаний, 2010. – 405 с.
- 8) Златопольский Д.М. Программирование: типовые задачи, алгоритмы, методы. – М.: Бином. Лаборатория знаний, 2007. – 223 с.
- 9) Кирюхин В.М., Цветкова М.С. Информатика. Программы внеурочной деятельности учащихся по подготовке к Всероссийской олимпиаде школьников: 5 – 11 классы М.: Бином. Лаборатория знаний, 2014.

6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП

Коньшин Иван Вячеславович, педагог дополнительного образования, преподаватель по направлению «Программирование на языке C++». Образование незаконченное высшее.

7. МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ПО РАЗДЕЛАМ ПРОГРАММЫ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Брату лучшую половину

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У Ани есть N братьев. В один из августовских дней Аня насобирала в саду много яблок. Аня была доброй и заботливой сестрой. Проходя мимо младшего брата, она отдала ему половину своих яблок и еще пол-яблока. Следующему по старшинству она также отдала половину оставшихся яблок и еще пол-яблока. И так она поступила со всеми своими братьями. Оказалось, что после того, как Аня отдала яблоки последнему брату, у нее самой яблок не осталось совсем. Кроме того, во время раздачи яблок не было разрезано ни одно из яблок.

Какое минимальное число яблок могло быть у Ани было первоначально, чтобы все эти условия оказались выполненными?

Формат входных данных

На вход подается единственное число N – количество братьев Ани ($0 \leq N < 32$).

Формат выходных данных

Требуется вывести одно натуральное число – минимально возможное изначальное число яблок у Ани, чтобы все условия задачи были выполнены.

Система оценки

При успешном прохождении тестов вам начисляется определенное количество баллов за каждый тест. Тестирование подзадачи начинается только в том случае, если пройдены все тесты в необходимых подзадачах.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
0	Баллы не начисляются	Тесты из условия	
1	25 тестов по 4 балла каждый	$0 \leq N < 32$	0

Пример

Стандартный ввод	Стандартный вывод
3	7

Школа менеджмента

Имя входного файла: input.txt или стандартный поток ввода
Имя выходного файла: output.txt или стандартный поток вывода
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Совсем скоро открывается школа "Менеджмент для самых маленьких". Всех юных менеджеров собираются учить в одной большой аудитории, где парты расставлены в R рядов, по C парт в каждом ряду, где за каждой партой есть K мест. Помогите организаторам школы менеджмента понять, подходит ли им эта аудитория. Требуется определить, влезут ли все учащиеся в данную аудиторию (на одном месте может сидеть только один человек), и можно ли рассадить учеников, из которых B мальчиков и G девочек, так, чтобы за одной партой не сидели одновременно и юные менеджеры мальчики, и юные менеджеры девочки.

Формат входного файла

В единственной строке даны 5 целых чисел: R, C, K, B, G ($1 \leq R, C, K \leq 103$, $0 \leq B, G \leq 109$).

Формат выходного файла

В первой строке требуется вывести YES, если мест в аудитории достаточно для всех учеников, и NO в противном случае. Если мест в аудитории достаточно, то во второй строке требуется вывести YES, если можно рассадить всех таким образом, чтобы мальчики и девочки не сидели за одной партой, и NO в противном случае.

Пример входных и выходных файлов

input.txt	output.txt
2 3 2 9 3	NO
2 3 2 9 3	YES NO

Описание системы оценивания

В этой задаче 20 тестов. Каждый тест оценивается в 5 баллов. Баллы за каждый тест начисляются независимо.

Растение

Имя входного файла: input.txt или стандартный поток ввода
Имя выходного файла: output.txt или стандартный поток вывода
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Растение вида «Информатикус» растёт по строгим математическим законам. Изначально оно имеет всего 1 лист. Если полить это растение удобрением «Добавляйка», на нём вырастает 4 новых листа, а если полить удобрением «Умножайка», то количество листьев увеличится ровно в 3 раза. Удобрение «Добавляйка» стоит X рублей, а «Умножайка» – Y рублей.

Алексею очень нравится число N, и он хочет, чтобы на его «Информатикусе» было ровно N листьев. Естественно, Алексей хочет потратить на это как можно меньше денег. Какое же минимальное количество рублей необходимо потратить на удобрения, чтобы получить растение, на котором будет ровно N листьев? Или, может быть, это вообще невозможно?

Формат входного файла

В единственной строке содержится три целых числа: X, Y, N, разделенных пробелами ($1 \leq X, Y \leq 100$, $1 \leq N \leq 1015$).

Формат выходного файла

Если получить ровно N листьев невозможно, вывести одну строчку «IMPOSSIBLE», иначе вывести одно целое число – минимальное количество рублей, которое нужно потратить на удобрения для получения растения, на котором будет ровно N листьев.

Примеры входных и выходных файлов

input.txt	output.txt
2 5 15	7
1 1 2	IMPOSSIBLE

Пояснение к примеру

В первом примере оптимальным решением является полив «Добавляйкой», а затем «Умножайкой», что требует $(2 + 5) = 7$ рублей.

Описание подзадач и системы оценивания

Баллы за каждый тест начисляются независимо от результатов других тестов.

Лошадью ходи, лошадью...

Имя входного файла: chevaux.in
Имя выходного файла: chevaux.out
Ограничения по времени: 2 секунды
Ограничения по памяти: 64 мегабайта

На шахматной доске, состоящей из M строк и N столбцов, размещены два шахматных коня — белый и чёрный. Каждый конь располагается в одной клетке, но возможна ситуация, когда в одной и той же клетке находятся оба коня.

Кони делают ходы по очереди в соответствии с правилами движения шахматного коня (первым ходит белый конь). Целью игры является как можно более быстрое размещение обоих коней в одной клетке.

Сможете ли вы рассчитать количество ходов, необходимое для завершения

игры, исходя из начального расположения фигур?

Формат входных данных.

Первая строка файла содержит величины M и N ($2 \leq M, N \leq 1000$). Во второй и третьей строке записаны координаты клеток, в которых находится соответственно белый и чёрный конь. Первая координата находится в границах от 1 до M , вторая — в границах от 1 до N .

Формат выходных данных.

Выведите единственное число — количество ходов, необходимое для завершения игры. Если кони никогда не смогут быть помещены в одну клетку, выведите -1 .

Пример входных и выходных данных

8 10	4
2 4	
7 9	

Пояснение. Конь может пойти на любое поле доски, если она располагается на другом конце русской буквы Г (то есть вначале конь перемещается на два поля по горизонтали или по вертикали, а затем на одну клетку перпендикулярно первоначальному направлению). Выходить за границы доски, естественно, нельзя.

Ров вокруг замка

Имя входного файла: ditch.in
Имя выходного файла: ditch.out
Ограничения по времени: 2 секунды
Ограничения по памяти: 64 мегабайта

Владелец старинного замка принимает в нём много туристов. Чтобы ещё больше увеличить их число, он решил восстановить давно засыпанный ров вокруг стен замка, доверху заполнить его водой и запустить в неё красивых рыб. Архитектор, которому поручена эта работа, убедил хозяина, что ров будет особенно красив, если его ширина будет одинаковой и равной A , а ближний к замку край рва будет отстоять от стен на одинаковое расстояние B .

Поскольку стоимость строительства рва зависит от площади поверхности воды, владелец попросил вас рассчитать эту площадь.

План замка представляет собой строго выпуклый многоугольник, а перепадами высот на месте строительства можно пренебречь.

Формат входных данных.

Первая строка содержит три целых числа: N — количество вершин многоугольника, образующего замок ($3 \leq N \leq 1000$), а также A и B ($1 \leq A, B \leq 100$). Каждая из последующих N строк содержит два целых числа, не превосходящих по модулю 10000 — координаты очередной вершины многоугольника. Вершины перечисляются в порядке их обхода против часовой

стрелки. Гарантируется, что многоугольник строго выпуклый.

Формат выходных данных.

Выведите единственное число — площадь поверхности воды, которой будет заполнен ров, абсолютная или относительная погрешность вашего ответа не должна превышать 10^{-4} .

Пример входных и выходных данных

4 5 2	228.20040098
0 0	
4 0	
7 4	
3 3	

Вложение денег

Имя входного файла: investment.in
Имя выходного файла: investment.out
Ограничения по времени: 2 секунды
Ограничения по памяти: 64 мегабайта

У вас неожиданно появились свободные деньги, и вы решили их выгодно вложить. Увы, за свою жизнь вы убедились, что для ведения собственного бизнеса у вас нет способностей, и вы решили вложить деньги в надёжные инвестиционные проекты.

Вы рассмотрели представленные заявки и выбрали N проектов, в которые можно вложить свои деньги без риска их потерять. Объём инвестиций в i -й проект ($1 \leq i \leq N$) составляет D_i , а доход, который вы рассчитываете получить после реализации этого проекта, равен T_i . Максимальный объём ваших инвестиций равен M .

К сожалению, все выбранные проекты начинаются одновременно, так что доход, полученный от одного проекта, инвестировать в другой нельзя... Кроме того, инвестировать в один проект более одного раза запрещается.

Определите максимальный доход, который вы сможете получить после реализации всех проектов, в которые вы вложили деньги.

Формат входных данных.

Первая строка содержит величины N и M ($1 \leq N \leq 1000$, $1 \leq M \leq 10^9$). Далее следуют N строк, описывающих каждый проект и содержащих величины D_i и T_i ($1 \leq D_i \leq 10^6$, $D_i < T_i \leq D_i + 100$).

Формат выходных данных.

Выведите единственное число — максимальную сумму полученного вами дохода (с учётом сделанных инвестиций).

Примеры входных и выходных данных

3 10 7 11 4 6 5 8	15
3 10 11 20 13 14 20 21	10