

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №46
С УГЛУБЛЕННЫМ ИЗУЧЕНИЕМ ОТДЕЛЬНЫХ ПРЕДМЕТОВ

ПРИНЯТА

на заседании педагогического
совета
протокол № 8
«09» апреля 2024 г.

Прямоугольник



УТВЕРЖДАЮ:

Директор МБОУ СОШ № 46 с

УИОП

/ Л.В. Гейнц/

«09» апреля 2024 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
(ОБЩЕРАЗВИВАЮЩАЯ) ПРОГРАММА**

технической направленности

«Юный программист»

(наименование программы)

Срок реализации программы: 1 год
Возраст обучающихся 13-17 лет
Количество часов в год: 76
Автор-составитель программы:
Никитина Н.Л., педагог дополнительного
образования

г. Сургут, 2024

АННОТАЦИЯ

Дополнительная общеобразовательная (общеразвивающая) программа «Юный программист», технической направленности рассчитана на учащихся 13-17 лет, срок реализации 1 год, объем программы 76 часов.

В процессе обучения учащиеся приобретут опыт разработки алгоритмов и программ на языке программирования, удовлетворяющих заданному описанию олимпиадной задачи; критического оценивания полученных результатов. Основным результатом обучения является пропаганда научных знаний и развитие у учащихся интереса к участию в олимпиадах по программированию, создание оптимальных условий для выявления одарённых и талантливых школьников, их дальнейшего интеллектуального роста и профессиональной ориентации.

Форма проведения занятий – очная с использованием электронного обучения. Под электронным обучением понимается организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации программы информации и обеспечивающих её обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу информации по линиям связи и автоматизированных серверов тестирования.

Основными формами организации деятельности на занятии являются групповые, индивидуально-групповые и индивидуальные занятия. Для успешного усвоения обучающимися данного курса предполагается применение фронтальных, групповых и индивидуальных приёмов работы, постепенный переход от работы со всей группой, через этап оказания дозированной помощи обучающемуся, к полностью самостоятельной работе.

Программа курса «Юный программист» предусматривает формирование у учащихся общеучебных умений и навыков, универсальных способов деятельности и ключевых компетенций. В этом направлении приоритетными для курса являются: основы и техника программирования и решения олимпиадных задач; реализация различных алгоритмов и умение применять их для решения вычислительных задач.

ПАСПОРТ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ (ОБЩЕРАЗВИВАЮЩЕЙ) ПРОГРАММЫ

Наименование образовательной организации Муниципальное бюджетное общеобразовательное учреждение средняя общеобразовательная школа №46 с углубленным изучением отдельных предметов

Название программы	Юный программист
Направленность программы	Техническая
Ф.И.О. педагога, реализующего дополнительную общеобразовательную программу	Никитина Нина Леонидовна
Год разработки	2024
Где, когда и кем утверждена дополнительная общеобразовательная программа	Принята на заседании педагогического совета, протокол № 8, «09» апреля 2024 г. Утверждена приказом от 27.04.2024 № ш46-13-516/4
Информация о наличии рецензии	-
Цель	Раскрытие значения программирования и сути профессии программиста, ознакомление учащихся со средой и основами программирования на языке С++, подготовка учащихся к практическому использованию полученных знаний при решении учебных задач, а в дальнейшем и в профессиональной деятельности, вовлечение учащихся в участие в олимпиадах по программированию разного уровня
Задачи	<p>Обучающие:</p> <ul style="list-style-type: none"> ✓ сформировать у учащихся системы знаний по структурному программированию, основным методам построения и анализа алгоритмов, средствам разработки, отладки и тестирования программ; ✓ углубить знания по программированию средствами языка программирования С++

	<ul style="list-style-type: none"> ✓ формировать опыт применения различных методов решения задач, реализуемых на языках C++. <p>Развивающие:</p> <ul style="list-style-type: none"> ✓ развивать навыки программирования алгоритмических структур; ✓ развивать логическое мышление учащихся; ✓ развивать интеллект учащихся. <p>Воспитательные:</p> <ul style="list-style-type: none"> ✓ учить планировать, готовить исправлять свои ошибки, вырабатывать культуру программирования; ✓ учить работать в группе, самостоятельно, дистанционно, вести дискуссию, корректно формулировать вопросы
<p>Ожидаемые результаты освоения программы</p>	<p><i>Личностные результаты:</i></p> <ul style="list-style-type: none"> ✓ подготовка учащихся к профессиональной деятельности, повышение уровня самооценки, уверенности в себе, выработка настойчивости; ✓ формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики; ✓ формирование коммуникативной компетентности в процессе образовательной, учебно-исследовательской, творческой и других видов деятельности. <p><i>Метапредметные результаты:</i></p> <ul style="list-style-type: none"> ✓ умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности; ✓ владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности; ✓ подготовка учащихся к практическому использованию полученных знаний при решении учебных задач. <p><i>Предметные результаты:</i></p> <ul style="list-style-type: none"> ✓ раскрытие значения программирования и сути профессии программист, ознакомление

	учащихся со средой и основами программирования на языке C++, участие в олимпиадах по программированию разного уровня
Срок реализации программы	2024-2025 уч. год
Количество часов в неделю / год	2 часа в неделю/76 часов в год
Возраст обучающихся	13-17 лет
Формы занятий	Видеолекции, практические занятия, дистанционные формы занятий, тестирование для самопроверки
Методическое обеспечение	Методическое обеспечение дополнительной общеобразовательной программы осуществляется посредством активного внедрения в образовательный процесс комплекса дидактических материалов: <ul style="list-style-type: none"> ✓ видео лекции, раскрывающие содержание каждой изучаемой темы, состоящие из двух частей; ✓ конспекты, где более подробно изложен лекционный материал в текстовом формате; ✓ вопросы и тесты для самопроверки; ✓ задачи с автоматической проверкой на образовательной платформе stepik.org
Условия реализации программы (оборудование, инвентарь, специальные помещения, ИКТ и др.)	<p><i>Средства обучения:</i></p> <ul style="list-style-type: none"> ✓ Автоматизированное рабочее место обучающегося с программным обеспечением, оборудованное в соответствии с санитарно-гигиеническими нормами; ✓ Демонстрационный экран (интерактивная доска); ✓ Локальная сеть; доступ к сети Интернет <p><i>Программное обеспечение:</i></p> <ul style="list-style-type: none"> ✓ Операционная система: Windows 10 ✓ Среда программирования C\C++ <p><i>Учебно-методическое обеспечение:</i></p> <ul style="list-style-type: none"> ✓ Дидактический материал в электронном виде

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Нормативно-правовое обеспечение программы

Программа разработана на основании законодательных и нормативно - правовых документов:

1. [Федеральный Закон Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»](#) (с изменениями).
2. [Распоряжение Правительства Российской Федерации от 31.03.2022 № 678-р «Об утверждении Концепция развития дополнительного образования детей до 2030 года»](#).
3. [Приказ Министерства просвещения РФ от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»](#).
4. [Постановление Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»](#).

А также другими Федеральными законами, иными нормативными правовыми актами РФ, законами и иными нормативными правовыми актами субъекта РФ (Ханты-Мансийского автономного округа – Югры), содержащими нормы, регулирующие отношения в сфере дополнительного образования детей, нормативными и уставными документами МБОУ СОШ №46 с углубленным изучением отдельных предметов.

Реализация дополнительной общеобразовательной (общеразвивающей) программы «Юный программист» осуществляется за пределами Федеральных государственных образовательных стандартов и не предусматривает подготовку обучающихся к прохождению государственной итоговой аттестации по образовательным программам.

До недавнего времени программирование составляло основной, а изначально – единственный раздел информатики. Совершенствование технологических и программных средств, привело к снижению количества часов, отводимых для изучения темы «Алгоритмизация и программирование» в программе среднего (полного) общего образования по информатике. Современные визуальные, мультимедийные пользовательские среды являются теми конкурентами, которые вытесняют разработку программ из сферы интересов учащихся. Для работы за компьютером, для поиска информации в сети пользователь имеет простые красивые инструменты, не требующие мыслительных усилий при применении.

В жизни человека алгоритмы встречаются повсеместно. Рассматривая алгоритмы как последовательность действий, можно выявить причинноследственные связи, и, таким образом, найти средства для достижения

искового результата. Часто именно алгоритм является методикой или технологией созидательного процесса. Многие языки программирования требуют однозначного описания последовательности операторов, то есть алгоритма. Между тем учащиеся, умеющие составить алгоритм и писать по нему программу, обычно значительно легче овладевают и пользовательскими навыками, так как понимают механизм управления компьютером. Они лучше успевают и по другим предметам, поскольку культура их мышления выше. В связи с этим представляется достаточно важным привить учащимся навыки алгоритмического мышления как можно более с раннего возраста.

Есть задача, проблема. Ученику требуется найти решение путем разработки соответствующей программы. Если решение известно, решались аналогичные задачи, то задействуется ассоциативная составляющая интеллекта, работа сводится к набору программы и ее отладке. Если решение неизвестно, то за постановкой задачи следует гипотеза и разработка первоначального варианта программы. Затем она подвергается исследованию, экспериментальной проверке с помощью системы тестов – сравнению ожидаемых результатов и полученных. Ученику мысленно следует предсказать, предвидеть результаты работы. Наступает фаза или экспериментального опровержения, или экспериментального подтверждения. Таким образом, деятельность при разработке программ характеризуется контролируемостью, обоснованностью и целенаправленностью. Оценка своих действий – неперенный атрибут программирования. На каждом шаге ученик имеет возможность осознать, насколько правильно принятое решение, насколько верен ход рассуждений, все ли факты учтены при принятии решения. Деятельность при программировании можно назвать направленной на получение желаемого результата. Она не просто активна, она сверхактивна, и мы видим возможность реализации концепции развивающего обучения в полном объеме. Кроме того, исследователи в области психологии информационных технологий считают, что подлинно квалифицированная работа с компьютером способствует избавлению ребенка от боязни допустить ошибку. Тем, кто обучался программированию, удается «избежать связи ошибки с неудачей». У учащихся появляется иное отношение к ошибкам: они перестают их бояться, умеют самостоятельно их устранить, и могут установить их источник. Все это создает дополнительные стимулы для творческих поисков нешаблонных путей решения задач, что так необходимо при решении задач по программированию олимпиадного уровня.

Решение олимпиадных задач позволяет раскрыть творческий потенциал учащихся во время подготовки к олимпиаде, учитывая возрастные особенности и перспективу развития личности. Использование многоуровневых олимпиадных задач, позволяет применить свой творческий потенциал, независимо от уровня подготовки.

Актуальность дополнительной общеобразовательной программы

Важной задачей образования является работа с одаренными учащимися, их подготовка к предметным олимпиадам. Олимпиада по информатике занимает одно из ведущих мест, в связи с интенсивным развитием информационных технологий.

Участие в олимпиадах позволяет развивать творческие способности учащихся и обеспечивает высокую мотивацию к образовательной деятельности.

Новизна программы

Данная программа отличается от существующих программ более углубленным изучением программирования; нововведениями в формах диагностики и подведения итогов реализации программы, выполняемые в формате олимпиад по программированию.

Направленность программы

Программа имеет техническую направленность, в связи с этим рассматриваются два актуальных аспекта изучения.

Технологический. Содержание программы рассматривается как средство формирования образовательного потенциала, позволяющего развивать наиболее передовые технологии – информационные, интегрирующие в себе науку, технологию, инженерное дело, математику.

Общеобразовательный. Содержание программы рассматривается как средство развития основных познавательных процессов, умения анализировать, выявлять сущности и отношения, описывать планы действий и делать логические выводы, опираясь на такие дисциплины, как механика, теория управления, схемотехника, программирование, теория информации, дискретная математика.

Уровень освоения программы

Уровень освоения программы – стартовый.

Дополнительная общеобразовательная программа предназначена для обучающихся 6-10 классов. Определяющим фактором является активное желание ученика.

В течение учебного года проводится мониторинг образовательного процесса, механизм отслеживания которого являются: учебная активность обучающихся на занятии, уровень их социальной активности и интерес к самостоятельной деятельности. Механизмами оценивания являются педагогическое наблюдение, стабильность групп и рефлексия. Занятия проходят в разновозрастных группах.

Отличительные особенности программы

Отличительные особенности данной программы заключаются в следующих приоритетных идеях: ориентация на личностные интересы, потребности, способности обучающегося; единство обучения, воспитания и развития; практико-деятельностная основа образовательного процесса, направленная на привлечение учащихся к выполнению технических задач.

Адресат программы

Программа ориентирована на учащихся 13-17 лет, обладающих повышенной мотивацией к изучению информатики и имеющих начальные знания в области алгоритмизации на уровне понимания простейших алгоритмов.

Количество обучающихся в группе: 15-20 человек.

Срок освоения программы: 1 год.

Объем программы: 76 часов.

Режим занятий: 2 раза в неделю.

Форма(ы) обучения: очная, в программе предусмотрены формы проведения занятий и формы контроля для работы в дистанционном режиме.

Особенности организации образовательного процесса

Данная программа позволяет провести непрерывную подготовку к олимпиадам по информатике, используя методическую коллекцию олимпиадных задач. В курсе использован системный подход при разработке разделов непрерывной подготовки одаренных детей к олимпиадам по информатике.

ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

Цель программы - раскрытие значения программирования и сути профессии программиста, ознакомление учащихся со средой и основами программирования на языке C++, подготовка учащихся к практическому использованию полученных знаний при решении учебных задач, а в дальнейшем и в профессиональной деятельности, вовлечение учащихся в участие в олимпиадах по программированию разного уровня.

Задачи:

Обучающие:

- ✓ сформировать у учащихся системы знаний по структурному программированию, основным методам построения и анализа алгоритмов, средствам разработки, отладки и тестирования программ;
- ✓ углубить знания по программированию средствами языка программирования C++
- ✓ формировать опыт применения различных методов решения задач, реализуемых на языках C++.

Развивающие:

- ✓ развивать навыки программирования алгоритмических структур;
- ✓ развивать логическое мышление учащихся;
- ✓ развивать интеллект учащихся.

Воспитательные:

- ✓ учить планировать, готовить исправлять свои ошибки, вырабатывать культуру программирования;
- ✓ учить работать в группе, самостоятельно, дистанционно, вести дискуссию, корректно формулировать вопросы.

УЧЕБНЫЙ ПЛАН

№	Название раздела	Количество часов			Формы аттестации / контроля
		всего	теория	практика	
1	Линейный алгоритм	6	3	3	тест, зачет: решение не менее 8 задач
2	Разветвляющие алгоритмы	6	3	3	тест, зачет: решение не менее 8 задач
3	Циклические алгоритмы	6	3	3	тест, зачет: решение не менее 9 задач
4	Задачи на закрепление алгоритмических конструкций	6	0	6	тест, зачет: решение не менее 9 задач
5	Массивы. Обработка массивов	6	2	4	тест, зачет: решение не менее 7 задач
6	Функции. Процедуры	4	2	2	тест, зачет: решение не менее 7 задач

7	Матрицы. Обработка матрицы	6	2	4	тест, зачет: решение не менее 14 задач
8	Строки. Обработка строк	6	2	4	тест, зачет: решение не менее 13 задач
9	Особенности решения олимпиадных задач	10	5	5	тест, зачет: решение не менее 14 задач
10	Повторение. Итоговое тестирование	10	2	8	тестирование
11	Решение и разбор олимпиадных задач	10	4	6	участие в ВОШ
Итого		76	28	48	

СОДЕРЖАНИЕ ПРОГРАММЫ

Тема 1. Линейный алгоритм (6 ч.)

Теория: Типы данных. Арифметические выражения. Арифметические операции и встроенные функции.

Практические работы: Типы данных. Арифметические выражения. Арифметические операции и встроенные функции. Решение сложных задач

Тема 2. Разветвляющие алгоритмы (6 ч.)

Теория: Разветвляющиеся алгоритмы с простыми условиями. Разветвляющиеся алгоритмы с составными условиями.

Практические работы: Разветвляющиеся алгоритмы с простыми условиями. Разветвляющиеся алгоритмы с составными условиями.

Тема 3. Циклические алгоритмы (6 ч.)

Теория: Циклические алгоритмы с параметром. Циклические алгоритмы с условием.

Практическая работа: Циклические алгоритмы с параметром. Циклические алгоритмы с условием. Решение сложных задач.

Тема 4. Задачи на закрепление алгоритмических конструкций (6 ч.)

Практическая работа: Продвинутое задачи на арифметические операции. Продвинутое задачи на условный оператор. Продвинутое задачи на циклы.

Тема 5. Массивы. Обработка массивов (6 ч.)

Теория: Массивы. Обработка массивов.

Практические работы: Массивы. Обработка массивов. Решение сложных задач.

Тема 6. Функции. Процедуры (4 ч.)

Теория: Функции. Процедуры

Практические работы: Функции. Процедуры

Тема 7. Матрицы. Обработка матрицы (6 ч.)

Теория: Матрицы. Обработка массивов.

Практические работы: Матрицы. Обработка массивов. Решение сложных задач.

Тема 8. Строки. Обработка строк (6 ч.)

Теория: Строки. Обработка строк.

Практические работы: Строки. Обработка строк. Решение сложных задач

Тема 9. Особенности решения олимпиадных задач (10 ч.)

Теория: Особенности работы с вещественными числами. Вектора. Указатели. Сложность алгоритмов. Работа с файлами.

Практические работы: Особенности работы с вещественными числами. Вектора. Указатели. Сложность алгоритмов. Работа с файлами.

Тема 10. Повторение. Итоговое тестирование (10 ч.)

Практические работы: Контрольный тест №1. Контрольный тест №2. Контрольный тест №3. Итоговый тест

Тема 11. Решение и разбор олимпиадных задач (10 ч.)

Практические работы: Разбор олимпиадных задач (школьный этап). Разбор олимпиадных задач (муниципальный этап)

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОГРАММЫ

Сформулированные цели реализуются через достижение образовательных результатов. Эти результаты структурированы по ключевым задачам общего образования, отражающим индивидуальные, общественные и государственные потребности, и включают в себя предметные, метапредметные и личностные результаты. Образовательные результаты сформулированы в деятельностной форме.

Личностные результаты:

- ✓ формирование ответственного отношения к учению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
- ✓ формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики;
- ✓ подготовка учащихся к профессиональной деятельности, повышение уровня самооценки, уверенности в себе, выработка настойчивости.

Метапредметные результаты:

- ✓ умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
- ✓ владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- ✓ умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
- ✓ умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- ✓ смысловое чтение;
- ✓ умение осознанно использовать речевые средства в соответствии с задачей коммуникации; владение устной и письменной речью;
- ✓ подготовка учащихся к практическому использованию полученных знаний при решении учебных задач.

Предметные результаты:

Знать:

- ✓ структуру программы на языке C++;

- ✓ основные типы данных языка C++;
- ✓ назначение и основные библиотеки функций C++;
- ✓ основные стандартные математические функции;
- ✓ функции ввода-вывода;
- ✓ спецификации формата ввода-вывода;
- ✓ логические операции;
- ✓ назначение условного оператора, его запись на языке C++;
- ✓ назначение оператора выбора, его запись на языке C++;
- ✓ особенности циклов с параметром, с предусловием, с постусловием;
- ✓ определение массива, индекса, размера, базового типа;
- ✓ способы инициализации массива;
- ✓ формат объявления, определения и вызова функций пользователя в C++;
- ✓ особенности локальных и глобальных переменных;
- ✓ определение рекурсии;
- ✓ определение указателя, динамической памяти;
- ✓ функции для работы с динамической памятью.

Умение:

- ✓ осуществлять ввод и вывод данных в программе;
- ✓ составлять программы линейной структуры для решения задач;
- ✓ использовать полную и неполную формы условного оператора;
- ✓ составлять программы разветвляющейся структуры для решения задач;
- ✓ составлять программы циклической структуры для решения задач;
- ✓ применять рекурсивные функции для решения задач;
- ✓ создавать одномерные динамические массивы;
- ✓ изменять размер одномерного динамического массива в ходе исполнения программы;
- ✓ составлять программы для решения задач с использованием динамической памяти.

Иметь практический опыт в:

- ✓ общих принципах разработки программ на языке программирования C++;
- ✓ участия в олимпиадах по программированию разного уровня.

Календарный учебный график

Количество учебных недель: 38

Количество учебных дней: 190

Сроки учебных периодов: 1 полугодие – 18; 2 полугодие – 20

№ п/п	Дата	Время проведения занятия	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля
Линейный алгоритм (6 ч.)							
1			Лекция.	1	Типы данных. Арифметические выражения	Кабинет информатики	Тест
2			Практическая работа	1	Практическая работа «Типы данных. Арифметические выражения»	Кабинет информатики	Зачет
3			Лекция	1	Арифметические операции и встроенные функции	Кабинет информатики	Тест
4			Практическая работа	1	Практическая работа «Арифметические операции и встроенные функции»	Кабинет информатики	Зачет
5			Практическая работа	1	Решение сложных задач	Кабинет информатики	Тест
6			Практическая работа	1	Решение сложных задач	Кабинет информатики	Зачет
Разветвляющие алгоритмы (6 ч.)							
7			Лекция	1	Разветвляющиеся алгоритмы с простыми условиями	Кабинет информатики	Тест
8			Практическая работа	1	Практическая работа «Разветвляющиеся алгоритмы с простыми условиями»	Кабинет информатики	Зачет

9			Лекция	1	Разветвляющиеся алгоритмы с составными условиями	Кабинет информатики	Тест
10			Практическая работа	1	Практическая работа «Разветвляющиеся алгоритмы с составными условиями»	Кабинет информатики	Зачет
11			Практическая работа	1	Решение сложных задач	Кабинет информатики	Тест
12			Практическая работа	1	Решение сложных задач	Кабинет информатики	Зачет
Циклические алгоритмы (6 ч.)							
13			Лекция	1	Циклические алгоритмы с параметром	Кабинет информатики	Тест
14			Практическая работа	1	Практическая работа «Циклические алгоритмы с параметром»	Кабинет информатики	Зачет
15			Лекция	1	Циклические алгоритмы с условием	Кабинет информатики	Тест
16			Практическая работа	1	Практическая работа «Циклические алгоритмы с условием»	Кабинет информатики	Зачет
17			Практическая работа	1	Решение сложных задач	Кабинет информатики	Тест
18			Практическая работа	1	Решение сложных задач	Кабинет информатики	Зачет
Задачи на закрепление алгоритмических конструкций (6 ч.)							
19			Лекция	1	Продвинутые задачи на арифметические операции	Кабинет информатики	Тест
20			Практическая работа	1	Практическая работа «Продвинутые	Кабинет информатики	Зачет

					задачи на арифметические операции»		
21			Лекция	1	Продвинутые задачи на условный оператор	Кабинет информатики	Тест
22			Практическая работа	1	Продвинутые задачи на условный оператор	Кабинет информатики	Зачет
23			Практическая работа	1	Продвинутые задачи на циклы	Кабинет информатики	Зачет
24			Практическая работа	1	Продвинутые задачи на циклы	Кабинет информатики	Зачет
Массивы. Обработка массивов (6 ч.)							
25			Лекция	1	Массивы	Кабинет информатики	Тест
26			Практическая работа	1	Массивы	Кабинет информатики	Зачет
27			Лекция	1	Обработка массивов	Кабинет информатики	Тест
28			Практическая работа	1	Обработка массивов	Кабинет информатики	Зачет
29			Практическая работа	1	Решение сложных задач	Кабинет информатики	Тест
30			Практическая работа	1	Решение сложных задач	Кабинет информатики	Зачет
Функции. Процедуры (4 ч.)							
31			Лекция	1	Функции	Кабинет информатики	Тест
32			Практическая работа	1	Функции	Кабинет информатики	Зачет
33			Лекция	1	Процедуры	Кабинет информатики	Тест
34			Практическая работа	1	Процедуры	Кабинет информатики	Зачет
Матрицы. Обработка матрицы (6 ч.)							

35			Лекция	1	Матрицы	Кабинет информатики	Тест
36			Практическая работа	1	Матрицы	Кабинет информатики	Зачет
37			Лекция	1	Обработка матрицы	Кабинет информатики	Тест
38			Практическая работа	1	Обработка матрицы	Кабинет информатики	Зачет
39			Практическая работа	1	Решение сложных задач	Кабинет информатики	Тест
40			Практическая работа	1	Решение сложных задач	Кабинет информатики	Зачет
Строки. Обработка строк (6 ч.)							
41			Лекция	1	Строки	Кабинет информатики	Тест
42			Практическая работа	1	Строки	Кабинет информатики	Зачет
43			Лекция	1	Обработка строк	Кабинет информатики	Тест
44			Практическая работа	1	Обработка строк	Кабинет информатики	Зачет
45			Практическая работа	1	Решение сложных задач	Кабинет информатики	Тест
46			Практическая работа	1	Решение сложных задач	Кабинет информатики	Зачет
Особенности решения олимпиадных задач (10 ч.)							
47			Лекция	1	Особенности работы с вещественными числами	Кабинет информатики	Тест
48			Практическая работа	1	Особенности работы с вещественными числами	Кабинет информатики	Зачет
49			Лекция	1	Вектора	Кабинет информатики	Тест
50			Практическая работа	1	Вектора	Кабинет информатики	Зачет

51			Лекция	1	Указатели	Кабинет информатики	Тест
52			Практическая работа	1	Указатели	Кабинет информатики	Зачет
53			Лекция	1	Сложность алгоритмов	Кабинет информатики	Тест
54			Практическая работа	1	Сложность алгоритмов	Кабинет информатики	Зачет
55			Лекция	1	Работа с файлами	Кабинет информатики	Тест
56			Практическая работа	1	Работа с файлами	Кабинет информатики	Зачет
Повторение. Итоговое тестирование (10 ч.)							
57			Тестирование	1	Контрольный тест №1	Кабинет информатики	Тест
58			Практическая работа	1	Практикум №1	Кабинет информатики	Зачет
59			Тестирование	1	Контрольный тест №2	Кабинет информатики	Тест
60			Практическая работа	1	Практикум №2	Кабинет информатики	Зачет
61			Тестирование	1	Контрольный тест №3	Кабинет информатики	Тест
62			Практическая работа	1	Практикум №3	Кабинет информатики	Зачет
63			Тестирование	1	Итоговый тест	Кабинет информатики	Тест
64			Тестирование	1	Итоговый тест	Кабинет информатики	Тест
65			Тестирование	1	Итоговый тест	Кабинет информатики	Тест
66			Тестирование	1	Итоговый тест	Кабинет информатики	Тест
Решение и разбор олимпиадных задач (10 ч.)							
67			Практическая работа	1	Разбор олимпиадных задач (школьный этап)	Кабинет информатики	Зачет

68			Практическая работа	1	Разбор олимпиадных задач (школьный этап)	Кабинет информатики	Зачет
69			Практическая работа	1	Разбор олимпиадных задач (школьный этап)	Кабинет информатики	Зачет
70			Практическая работа	1	Разбор олимпиадных задач (школьный этап)	Кабинет информатики	Зачет
71			Практическая работа	1	Разбор олимпиадных задач (школьный этап)	Кабинет информатики	Зачет
72			Практическая работа	1	Разбор олимпиадных задач (муниципальный этап)	Кабинет информатики	Зачет
73			Практическая работа	1	Разбор олимпиадных задач (муниципальный этап)	Кабинет информатики	Зачет
74			Практическая работа	1	Разбор олимпиадных задач (муниципальный этап)	Кабинет информатики	Зачет
75			Практическая работа	1	Разбор олимпиадных задач (муниципальный этап)	Кабинет информатики	Зачет
76			Практическая работа	1	Разбор олимпиадных задач (муниципальный этап)	Кабинет информатики	Зачет

Условия реализации программы

Средства обучения:

- ✓ Автоматизированное рабочее место обучающегося с программным обеспечением, оборудованное в соответствии с санитарно-гигиеническими нормами.
- ✓ Демонстрационный экран (интерактивная доска)
- ✓ Локальная сеть
- ✓ Доступ к сети Интернет

Программное обеспечение:

- ✓ Операционная система: Windows 10
- ✓ Среда программирования C\C++

Учебно-методическое обеспечение:

- ✓ Дидактический материал в электронном виде

Методическое обеспечение программы

При реализации данной программы используются информационно – коммуникационная, технологии проблемного и интегрированного обучения.

Процесс обучения по данной дополнительной общеобразовательной программе выстроен по классическим принципам дидактики: принципам научности, сознательности и самостоятельности, систематичности, последовательности, наглядности, доступности, связи теории с практикой.

Основными формами организации деятельности на занятии являются групповые, индивидуально-групповые и индивидуальные занятия. Для успешного усвоения обучающимися данного курса предполагается применение фронтальных, групповых и индивидуальных приёмов работы, постепенный переход от работы со всей группой, через этап оказания дозированной помощи обучающемуся, к полностью самостоятельной работе.

Используемые формы учебных занятий: лекция, видео лекция, конспекты, где более подробно изложен лекционный материал в текстовом формате; практикумы по программированию, дистанционная форма занятий, зачетная работа.

Для осуществления контроля уровня знаний, умений и навыков используются следующие методы: самоконтроль, индивидуальный устный опрос, практические работы, зачеты. По окончании изучения каждого раздела обучающийся должен иметь законченные программы.

Для обучения с применением дистанционных образовательных технологий используются электронные образовательные ресурсы на платформе stepik.org или при помощи средств виртуального общения «учитель - ученики» в «Сферум».

Воспитательный компонент

Основным результатом обучения является пропаганда научных знаний и развитие у школьников интереса к участию в олимпиадах по программированию высокого уровня, создание оптимальных условий для выявления одарённых и талантливых школьников, их дальнейшего интеллектуального роста и профессиональной ориентации. Учить планировать, разрабатывать программы, исправлять свои ошибки. Вырабатывать культуру программирования, учить работать

группе, самостоятельно, дистанционно, вести дискуссию, корректно формулировать вопросы.

Формы промежуточной аттестации и итогового контроля

Программа курса предусматривает использование зачетной системы как одной из форм контроля знаний учащихся, которая отличается от стандартных форм контроля характером проведения, системой оценивания. Зачет – это специальный этап контроля, целью которого является проверка достижения обучающимися уровня обязательной подготовки.

Зачетная система:

- ✓ организует и дисциплинирует учащихся;
- ✓ помогает систематизировать знания;
- ✓ повысить ответственность учащихся за результаты своего обучения;
- ✓ развить у учащихся самостоятельность мышления и способность к самообразованию и саморазвитию;
- ✓ обеспечить индивидуальный подход к каждому ученику;
- ✓ проявляет творческую активность;
- ✓ готовит к обучению в высших и средних учебных заведениях;
- ✓ снижает стресс при сдаче экзаменов.

Определяется число зачетных работ, которые учащийся должен сдать, несмотря на болезнь, отсутствие на занятии по другим причинам. Зачет за полугодие выставляется, если учащийся сдал все зачетные работы. Зачет за год выставляется при наличии зачетов за каждое полугодие.

Раздел	Количество задач	Условие выставления зачета (минимальное количество верно решенных задач)
Линейный алгоритм	12	8
Разветвляющиеся алгоритмы	13	8
Циклические алгоритмы	14	9
Задачи на закрепление	15	9
Массивы	11	7
Функции. Процедуры	11	7
Матрицы	22	14
Строки. Обработка строк	21	13

Зачетная система является механизмом ликвидации пробелов в знаниях учащихся. Она позволяет ученикам осваивать программу в разном темпе и на разных уровнях.

СПИСОК ЛИТЕРАТУРЫ

Для педагога:

1. Порублев И.Н., Ставровский А.Б. Алгоритмы и программы. Решение олимпиадных задач – М.: ООО «И.Д. Вильямс», 2017. – 480 с.: ил;
2. Информатика. Программирование : 10-11-й классы : учебное пособие : в 2-х частях / К.Ю. Поляков. – 2-е изд., стер. – Москва : Просвещение, 2022. – (Инженеры будущего) Ч. 1. – 208 с. : ил.
3. Информатика. Программирование : 10-11-й классы : учебное пособие : в 2-х частях / К.Ю. Поляков. – 2-е изд., стер. – Москва : Просвещение, 2022. – (Инженеры будущего) Ч. 2. – 192 с. : ил.
4. Козлов С.В., Быков А.А., Статья «Особенности обучения школьников олимпиадному программированию с использованием математического моделирования» // Современные наукоемкие технологии. – 2022. – № 6. – С. 141-146; URL: <https://top-technologies.ru/ru/article/view?id=39215> (дата обращения: 24.04.2024).
5. Глеб Евстропов, Статья «Чек-лист: олимпиадное программирование — с чего начать школьнику?», ведущий разработчик «Яндекса» и руководитель службы поисковых подсказок <https://rb.ru/young/chek-list-olimpiadnoe-programmirovanie/> (дата обращения: 24.04.2024).

Для обучающихся:

1. Программирование. Python. C++ : учебное пособие : в 4 частях / К.Ю. Поляков. – 4-е изд., стер. – Москва : Просвещение, 2022. – (Профильная школа). Ч.1-144 с. : ил.
2. Программирование. Python. C++ : учебное пособие : в 4 частях / К.Ю. Поляков. – 4-е изд., стер. – Москва : Просвещение, 2022. – (Профильная школа). Ч.2-176 с. : ил.
3. Программирование. Python. C++ : учебное пособие : в 4 частях / К.Ю. Поляков. – 4-е изд., стер. – Москва : Просвещение, 2022. – (Профильная школа). Ч.3-134 с. : ил.
4. Программирование. Python. C++ : учебное пособие : в 4 частях / К.Ю. Поляков. – 4-е изд., стер. – Москва : Просвещение, 2022. – (Профильная школа). Ч.4-192 с. : ил.

Интернет-источники:

Интернет-ресурсы для теоретической подготовки к олимпиадам:

1. <http://www.intuit.ru/courses.html> (сайт Интернет-университета информационных технологий);
2. <http://www.olympiads.ru/sng/index.shtml> (сайт МИОО, МЦНМО, и оргкомитета Московской олимпиады по информатике для проведения дистанционных семинаров по подготовке к олимпиадам по информатике);
3. <http://vzshit.net.ru/> (сайт Всесибирской заочной школы информационных технологий).

Интернет-ресурсы с коллекциями олимпиадных задач:

1. <http://old.info.rosolymp.ru> (сайт с самой большой в России коллекцией задач международных и всероссийских олимпиад по информатике с методическими рекомендациями по их решению);
2. <http://www.olympiads.ru/moscow/index.shtml> (сайт московских олимпиад по информатике);
3. <http://neerc.ifmo.ru/school/russia-team/archive.html> (сайт с архивом задач Всероссийских командных олимпиад школьников по программированию);
4. <http://contest.ur.ru> (сайт Уральских олимпиад по информатике);
5. <http://www.olympiads.ru/> (сайт по олимпиадной информатике);
6. <http://olimpic.nsu.ru/nsu/> (сайт открытой Всесибирской олимпиады по программированию им. И.В. Потгосина).

Интернет-ресурсы с коллекциями олимпиадных задач и возможностью их тестирования в реальном масштабе времени:

1. <http://acm.timus.ru/> (сайт Уральского государственного университета, содержащий большой архив задач с различных соревнований по спортивному программированию);
2. <http://acm.sgu.ru> (сайт Саратовского государственного университета, содержащий архив задач с системой онлайн-проверки).

Сайты интернет-олимпиад для учащихся:

1. <http://info-online.rusolimp.ru/> (сайт интернет-туров заключительного этапа Всероссийской олимпиады школьников по информатике);
2. <http://olymp.ifmo.ru/> (сайт городских интернет – олимпиад школьников Санкт-Петербурга);
3. <http://neerc.ifmo.ru/school/io/index.html> (сайт интернет-олимпиад по информатике, проводимых жюри Всероссийской командной олимпиады школьников по программированию);
4. <http://www.olympiads.ru/online/index.shtml> (сайт московских онлайн-олимпиад);
5. <http://olimpic.nsu.ru/acmSchool/archive/2006-2007/train2006/index.shtml> (сайт тренировочных олимпиад школьников, поддерживаемый Новосибирским государственным университетом)

На вход программе подаются два целых числа n , m каждое в отдельной строке $0 < n \leq 12$, $0 \leq m < 60$, указывающие момент времени "n часов m минут".

Определите наименьшее число полных минут, которое должно пройти до того момента, когда часовая и минутная стрелки на циферблате совпадут, не обязательно на каком-то делении. Вещественную арифметику не использовать.

Sample Input:

2 50

Sample Output:

26

Рис. 1. Пример сложной задачи по теме «Линейные алгоритмы»

Требуется определить, можно ли от шоколадки размером $n \times m$ долек отломить k долек, если разрешается сделать один разлом по прямой между дольками (то есть разломить шоколадку на два прямоугольника).

Входные данные

Вводятся 3 числа: n , m и k ; k не равно $n \times m$. Гарантируется, что количество долек в шоколадке не превосходит 30000.

Выходные данные

Программа должна вывести слово YES, если возможно отломить указанное число долек, в противном случае вывести слово NO.

Sample Input:

4
2
6

Sample Output:

YES

Рис. 2. Пример сложной задачи по теме «Разветвляющиеся алгоритмы»

Натуральное число называется числом Армстронга, если сумма цифр числа, возведенных в K -ю степень (где K – количество цифр в числе) равна самому числу.

Например, $153 = 1^3 + 5^3 + 3^3$.

Напишите программу, которая находит все числа Армстронга на отрезке $[a, b]$.

Входные данные

Входная строка содержит два натуральных числа – значения a и b , разделённых пробелами. Гарантируется, что $a \leq b$.

Выходные данные

Программа должна вывести в одну строчку все числа Армстронга на отрезке $[a, b]$, разделив их пробелами. Если таких чисел нет, программа должна вывести число -1.

Sample Input:

100 400

Sample Output:

153 370 371

Рис. 3. Пример сложной задачи по теме «Циклические алгоритмы»

Строки в книге

В книге на одной странице помещается K строк. Таким образом, на 1-й странице печатаются строки с 1-й по K -ю, на второй — с $(K+1)$ -й по $(2K)$ -ю и т.д. Напишите программу, которая по номеру строки в тексте определяет номер страницы, на которой будет напечатана эта строка, и порядковый номер этой строки на странице.

Входные данные

Вводятся два числа: K — количество строк, которое печатается на странице, и N — номер строки ($1 \leq K \leq 200$, $1 \leq N \leq 20000$).

Выходные данные

Выведите два числа — номер страницы, на которой будет напечатана эта строка, и номер строки на странице.

Sample Input:

50 1

Sample Output:

1 1

Рис. 4. Пример продвинутой задачи

Шеренга

Петя впервые пришел на урок физкультуры в новой школе. Перед началом урока ученики выстраиваются по росту, в порядке невозрастания. Напишите программу, которая определит на какое место в шеренге Пете нужно встать, чтобы не нарушить традицию, если заранее известен рост каждого ученика и эти данные уже расположены по невозрастанию (то есть каждое следующее число не больше предыдущего). Если в классе есть несколько учеников с таким же ростом, как у Пети, то программа должна расположить его после них.

Входные данные

Сначала задано число N — количество учеников (не считая Петю) ($1 \leq N \leq 1001 \leq N \leq 100$). Далее через пробел записаны N чисел — элементы массива. Массив состоит из натуральных чисел, не превосходящих 200 (рост учеников в сантиметрах). Затем, на новой строке, вводится рост самого Пети.

Выходные данные

Необходимо вывести единственное число - номер Пети в шеренге учеников.

Sample Input:

8
165 163 160 160 157 157 155 154
162

Sample Output:

3

Рис. 5. Пример сложной задачи по теме «Массивы»

Транспонирование матрицы

Напишите программу, которая выполняет транспонирование матрицы — преобразование, в результате которого строки становятся столбцами, а столбцы — строками.

Входные данные

В первой строке записаны через пробел размеры матрицы: количество строк N и количество столбцов M ($1 \leq N, M \leq 100$). В следующих N строках записаны строки матрицы, в каждой — по M натуральных чисел, разделённых пробелами.

Выходные данные

Программа должна вывести матрицу, полученную в результате транспонирования по строкам.

Sample Input:

4 5
1 2 3 4 5
6 7 8 9 3
5 4 3 2 1
7 9 8 7 6

Рис.6. Пример сложной задачи по теме «Матрицы»

Шифр Юлия

Юлий Цезарь использовал свой способ шифрования текста. Каждая буква заменялась на следующую по алфавиту через K позиций по кругу. Необходимо по заданной шифровке определить исходный текст.

Входные данные

В первой строке дана шифровка, состоящая из заглавных латинских букв. Во второй строке число K ($1 \leq K \leq 10$).

Выходные данные

Требуется вывести результат расшифровки.

Sample Input:

XPSE

1

Sample Output:

WORD

Рис.7. Пример сложной задачи по теме «Строки»

Протокол матча

В последнее время в одной из школ Н. Новгорода, а также на одном из факультетов ННГУ стала очень популярна игра в настольный теннис. Игроки часто сталкиваются со следующей проблемой: довольно трудно уследить за всем ходом матча и при этом не сбиться со счёта, поэтому очень хотелось бы иметь программу, подсчитывающую счёт. Напишите программу, которая по данному протоколу матча восстановит итоговый счёт.

Протокол состоит из последовательности следующих событий: *service*, *net*, *out*, *goal*, *return*, *eam*.

События обозначают следующее:

- * *service* — подача (при этом игрок ударяет по мячу). *service* — всегда первое событие во входном файле. После него могут следовать *net*, *out*, *goal*, *return*.
- * *net* — мяч ударяется о половину поля того игрока, который ударял по мячу последним, слишком много раз. Игрок, который ударял по мячу последним, проигрывает розыгрыш. После этого события могут идти *service* или *eam*.
- * *out* — мяч уходит в аут. Игрок, который ударял по мячу последним, проигрывает розыгрыш. После этого события могут идти *service* или *eam*.
- * *goal* — игрок, который ударял по мячу последним, забивает гол (т.е. выигрывает розыгрыш). Далее может быть *service* или *eam*.
- * *return* — игрок отбивает мяч, ударя по нему (игроки ударяют по мячу по очереди). Далее может быть *net*, *out*, *goal*, *return*.
- * *eam* — матч окончен. Это всегда последнее событие.

Когда игрок выигрывает розыгрыш, ему начисляется очко. Когда игрок проигрывает розыгрыш, очко начисляется его противнику.

Поддачи подаются по пять штук, т.е. первые пять подач подаёт первый игрок, следующие пять — другой и т.д. Полное количество подач может быть не кратным пяти, в таком случае последняя серия подач будет короче пяти штук.

Конечно, в реальном матче может произойти ситуация, которую невозможно описать этими событиями, но ваша программа должна считать, что весь матч описывается данными во входном файле событиями.

Входные данные

Во входном файле находится список событий. События расположены по одному на строке без пробелов. Последовательность событий удовлетворяет всему, что было сказано выше; пустых строк во входном файле нет (кроме, возможно, строк после события *eam*). Всего событий не более 50000.

Выходные данные

В выходной файл выведите два числа: очки того, кто подавал первым, потом — очки его противника.

Sample Input:

```
service
goal
service
out
service
net
service
return
return
return
out
service
return
goal
service
goal
eam
```

Sample Output:

```
2 4
```

Рис.7. Пример задачи для итоговой аттестации