

Общая информация

Продолжительность экзамена: **5 часов** (300 минут)

Формат проведения: Один компьютерный тур. На туре участнику предоставляется компьютер и предлагается решить **от 10 до 20 задач**. Решения участника проверяются автоматической тестирующей системой. При проверке решений нет человеческого фактора, поэтому апелляция не предусмотрена.

Языки программирования

Задачи можно сдавать только на двух языках программирования – C++ и Python 3. Разные задачи можно сдавать на разных языках программирования.

Комплект задач экзамена обеспечит небольшое преимущество тем, кто знает оба языка:

- Некоторые задачи (**не более 15%** от общего числа задач) можно сдать только на C++ (программы на Python работают примерно в 100 раз дольше аналогичных программ на C++ и не уложатся в ограничения по времени)
- Некоторые задачи (**не более 15%** от общего числа задач) существенно проще решать на Python. Само собой, эти задачи можно решить и на C++, однако решение будет существенно более объемным, а процесс его написания будет существенно более трудозатратным.

Алгоритмы и структуры данных

Опыт участия в олимпиадах (даже в школьном и муниципальном этапах ВсОШ) может обеспечить небольшое преимущество. Для решения некоторых задач необходимо будет применить алгоритмы или использовать структуры данных (полный список ниже). Однако таких задач будет **не более 25%**.

Таким образом, **75% задач экзамена будут на реализацию** («напишите то, что написано в условии»).

Философия экзамена

Мы стремимся набрать в математико-программистский профиль ребят, которые уверенно владеют одним из языков программирования (Python 3 или C++). В случае успешного поступления мы будем рекомендовать «подтянуть» уровень владения вторым языком программирования к началу учебного года. Алгоритмам и структурам данных мы научим уже после поступления в процессе обучения.

В связи с этим мы рассчитываем, что поступить на математико-программистский профиль можно будет и без знания алгоритмов и структур данных, уверенно владея только одним языком программирования (в этом случае участнику будут «доступны» как минимум 60% задач).

Рекомендуемый уровень владения языком программирования

Общие конструкции языков Python и C++

- Типы данных
- Ввод-вывод данных
- Условный оператор
- Циклы for и while
- Символы и строки
- Функции
- Работа с файлами (чтение из файла и запись в файл)
- Считывание до конца входных данных (неизвестное количество чисел и/или строк)

Особенности Python

- Списки
- Вложенные списки и матрицы
- Словари
- Множества
- Кортежи
- Сортировка списка кортежей или списка списков

Особенности C++

- Массивы / динамические массивы (векторы - `std::vector`)
- Двумерные массивы / массивы векторов
- Ассоциативные контейнеры `std::map` и `std::unordered_map`
- Контейнеры `std::set` и `std::unordered_set` (хранят только ключи, без ассоциированных значений)
- Структуры
- Сортировка массива структур

Алгоритмы и структуры данных

- Sum, Count, Min и Max на префиксе и/или суффиксе
- Метод двух указателей
- Бинарный поиск
- Бинарный поиск по ответу
- Динамическое программирование
- Очередь и стек

Применение перечисленных выше алгоритмов и структур данных потребуется **не более, чем в 25% задач**.

Мы рассчитываем, что поступить на математико-программистский профиль можно будет и без знания алгоритмов и структур данных (т.е. решая остальные 75% задач).

Технические особенности проведения экзамена

1. Для решения задач участники могут использовать следующие языки программирования и соответствующие им среды разработки

C++	▪ Visual Studio Community 2022 ▪ Visual Studio Code ▪ Code::Blocks
Python 3	▪ PyCharm Community Edition ▪ Wing 101 ▪ Wing Personal ▪ Python 3 IDLE ▪ Visual Studio Code

2. **Использование любых онлайн-компиляторов и онлайн сред разработки запрещено** и приведет к удалению с экзамена с результатом «незачёт без права пересдачи»

3. Запрещается использовать любую информацию, размещенную в сети Интернет (**в том числе документацию по языкам программирования**). Запрещается пользоваться любой литературой и личными записями. Бумагу для черновиков выдадут организаторы экзамена.

4. **Запрещается иметь при себе** и использовать личные компьютеры, телефоны, мобильные средства связи и любые другие электронные устройства. В частности, запрещены любые наушники и любые калькуляторы. Отметим, что во время экзамена запрещается иметь при себе любые часы (даже механические).

5. Во время экзамена участники решают предложенные задачи. Решением задачи является программа, написанная на одном из допустимых языков программирования (Python 3 или C++). Программа не должна содержать вспомогательных модулей или файлов. **Разные задачи можно решать на разных языках программирования.**

6. Решения участников по каждой задаче проверяются на заранее подготовленном наборе тестов. Решение принимается (считается правильным), если оно прошло все тесты. **Частичные решения (прошедшие не все тесты) не оцениваются.**

7. Время тестирования и доступная память на каждом тесте ограничены. Решения, превысившие установленное ограничение, считаются неэффективными для данной задачи. В этом случае тест считается не пройденным, а решение, как следствие, неверным. Ограничения на время работы программы на одном тесте и на размер доступной памяти указываются в формулировках задач.

8. По мере готовности своих решений участник посылает их в тестирующую систему для проверки. Проверка решений производится автоматической

тестирующей системой прямо во время экзамена (проверка занимает около 1-5 минут).

9. Возможные вердикты тестирующей системы (т.е. результаты проверки решения участника):

Результат	Номер теста	Комментарий	Возможные причины
- Полное решение - Accepted - OK	Нет	Решение принято	Решение правильное
- Ошибка компиляции - Compilation error	Нет	В результате компиляции программы не создан исполняемый файл	- Синтаксическая ошибка в программе - При отправке на проверку выбран неверный язык программирования
- Неправильный ответ - Wrong answer	Да	Ответ неверен	- Неверный алгоритм решения - Неверный формат вывода
- Превышено ограничение времени - Time limit exceeded	Да	Программа не завершилась за отведенный период времени	- Неэффективное решение - Ошибка в программе
- Превышено ограничение памяти - Memory limit exceeded	Да	Программа попыталась использовать больше памяти, чем разрешается	- Неэффективное решение - Ошибка в программе
- Ошибка исполнения - Runtime error	Да	Программа завершилась с ненулевым кодом возврата, либо создала исключительную ситуацию (exсerption) и не обработала ее	- Ошибка времени исполнения - Необработанная исключительная ситуация - Деление на 0 - Бесконечная рекурсия - Обращение к несуществующему элементу строки / массива (выход за границы) “return (не 0)” в программе на C++ - В конце программы на C++ нет оператора “return 0”

Материалы для подготовки

Часть 1. Онлайн-курсы (теория + интерактивные задачи)

Язык программирования C++

- Введение в программирование на языке C++
<https://edu.sirius.online/course/cppintro>
- Введение в программирование (C++)
<https://stepik.org/course/363/promo>
- Основы C/C++ для спортивного программирования
<https://stepik.org/course/80538/promo>
- Яндекс Образование, хендбук «Основы C++»
<https://education.yandex.ru/handbook/cpp>

Язык программирования Python

- Введение в программирование на языке Python
<https://edu.sirius.online/course/pythonintro>
- "Поколение Python": курс для начинающих
<https://stepik.org/course/58852/promo>
- "Поколение Python": курс для продвинутых
<https://stepik.org/course/68343/promo>
- Яндекс Образование, хендбук «Основы Python»
<https://education.yandex.ru/handbook/python>

Алгоритмы и структуры данных

- Введение в алгоритмы: реализация на языке C++
<https://edu.sirius.online/course/algocpp>
- Введение в алгоритмы: реализация на языке Python
<https://edu.sirius.online/course/algopython>
- Яндекс Образование, хендбук «Основы алгоритмов»
<https://education.yandex.ru/handbook/algorithms>
- Алгоритмика
<https://ru.algorithmica.org/>

Материалы для подготовки

Часть 2. Архивы задач

Сайт проекта "Школа программиста"

<https://acmp.ru/>

Большое число задач подходящего уровня сложности. Страница «Архив задач» содержит очень удобный фильтр по темам.

Codeforces

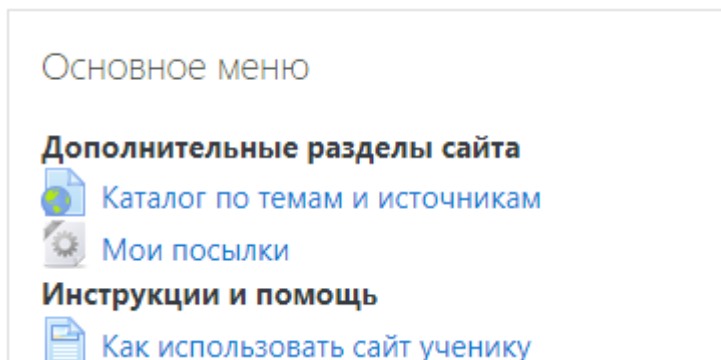
<https://codeforces.com/>

Огромный архив задач. Очень много задач на любую тему. Для эффективного поиска необходимо на странице «Архив» использовать «Фильтр задач» и возможность «Добавить тег» (т.е. «тему» задачи).

Информатикс

<https://informatics.msk.ru/>

Поиск задач: «Основное меню» - «Дополнительные разделы сайта» - «Каталог по темам и источникам»



Решение задачи: «К задаче №» - «Перейти»

Можно использовать готовые подборки задач, т.е. «Курсы».

Например, «Задачи из учебника К.Ю. Полякова и Е.А. Еремина»:

<https://informatics.msk.ru/course/view.php?id=666>

Timus Online Judge

<https://acm.timus.ru/>

Для решения многих задач необходимо применять алгоритмы и структуры данных (т.е. большая часть задач сложнее тех, что будут на вступительном экзамене)

Материалы для подготовки

Часть 3. Примеры задач, подходящих под формат экзамена

1. Пример простой задачи на работу с одномерными массивами (в C++) или списками (в Python). Для решения задачи необходимо написать простейший перебор (перебираем, над каким участком создаем искусственный дождь):

<https://codeforces.com/problemset/problem/66/B>

2. Пример простой задачи на работу с двумерными массивами (в C++) или с вложенными списками (в Python):

<https://acm.timus.ru/problem.aspx?num=1319>

3. Пример несложной задачи на реализацию, которая достаточно просто сдаётся на языке C++, однако на языке Python её решить существенно сложнее.

«Наивное решение» (просто написать то, что описано в условии) без проблем «зайдёт» на C++. Однако это же решение, написанное на языке Python, не уложится в ограничение по времени (1 секунда на каждый тест).

<https://codeforces.com/contest/585/problem/A>

4. Пример задачи на работу со строками, для решения которой необходимо применить метод динамического программирования:

https://acmp.ru/index.asp?main=task&id_task=865

5. Пример достаточно серьезной и сложной задачи на обработку строк. Задача на реализацию, т.е. нужно реализовать то, что описано в условии.

https://acmp.ru/index.asp?main=task&id_task=442

Задача прекрасно иллюстрирует принцип «тесты / примеры – неотъемлемая часть условия задачи». Некоторые тонкости могут быть непонятны после прочтения условия, однако тестовый пример их разъясняет.

Для решения задачи необходимо уметь считывать произвольное количество строк (до конца файла / до конца входных данных).

Знание простейшей структуры данных – стека существенно упростит решение задачи. Однако ограничения на длину HTML-документа позволяют решить задачу и без использования стека (решение квадратичной сложности)