

1 этап

Методика проведения испытания

Вступительное испытание 1-го этапа проводится очно в Университетском Лицее №1511 или №1523 в соответствии со списками распределения (становятся известны накануне даты соответствующего вступительного испытания).

В течение 60 минут абитуриент письменно решает вариант, далее проводится собеседование с преподавателем кафедры информатики Университетского Лицея №1511 или №1523, либо профильным сотрудником из НИЯУ МИФИ, в ходе которого производится оценка решений задач от 0 баллов до указанного максимально возможного балла за задание (по аналогии, как в демоварианте). Максимальный балл за выполнение испытания 1-го этапа — 100 баллов, и вычисляется как сумма полученных баллов за все оценённые собеседующим задания.

Во время написания экзамена нельзя пользоваться любыми справочными материалами, гаджетами, смартчасами, нейросетями, средствами аудио и видеосвязи, переговариваться с остальными абитуриентами. В случае нарушения данных правил вы можете быть аннулированы и удалены с прохождения испытания.

Перечень разделов и тем для подготовки

Системы счисления.

Позиционные системы счисления, основание системы, разряды. Представление чисел в двоичной, восьмеричной, шестнадцатеричной и десятичной системах. Перевод чисел между системами счисления. Двоичное сложение и вычитание. Количество единиц в двоичной записи числа. Анализ выражений, содержащих степени двойки.

Кодирование.

Кодирование символов, таблицы соответствий, алфавиты. Однозначная и неоднозначная расшифровка. Понятие длины кода, использование таблиц символов. Принципы простейших шифров заменой. Решение задач на определение числа возможных расшифровок кодовых последовательностей.

Информационные модели. Путь в графе. Табличное представление информации.

Графы как модели объектов и связей. Вершины, рёбра, направления. Пути в графе, ограничения на пути. Подсчёт количества возможных путей по правилам. Табличная форма представления информации. Заполнение и анализ таблиц и схем. Информационные модели на основе таблиц соответствия, правил построения последовательностей.

Алгоритмы.

Понятие алгоритма. Последовательность команд. Линейные, разветвляющиеся, циклические конструкции. Порядок выполнения команд. Анализ простейших вычислительных алгоритмов. Построение алгоритмов в словесной форме и по схеме.

Исполнители.

Исполнитель на клетчатой плоскости. Команды перемещения: вверх, вниз, влево, вправо. Понятие стены и свободной клетки. Условия свободы соседних клеток. Цикл «ПОКА» и правила его выпол-

нения. Определение конечного результата программы исполнителя. Анализ поведения в ограниченном лабиринте.

Передача информации.

Информационные процессы: передача, хранение, обработка. Скорость передачи информации, единицы измерения информации. Простейшие модели передачи данных. Решение задач на объём информации и скорость передачи.

Анализ программы.

Трассировка — пошаговое выполнение инструкций. Переменные, присваивание. Арифметические операции в алгоритмах. Порядок вычислений, изменение значений переменных. Определение результата работы алгоритма по его тексту.

Литература и интернет-источники для подготовки

1. Босова, Л. Л. **Информатика. 8–9 классы : учебник для общеобразоват. организаций** / Л. Л. Босова, А. Ю. Босова. — М. : БИНОМ. Лаборатория знаний, 2023. — 304 с.
2. Лаптев, В. В. **Информатика. Теория и практика для школьников. 8–9 классы : учебное пособие** / В. В. Лаптев, А. А. Кузнецов. — СПб. : Питер, 2022. — 256 с.
3. Поляков, К. Ю. **Информатика. 8–9 классы : учебник для общеобразоват. организаций** / К. Ю. Поляков, Е. А. Еремин. — СПб. : Питер, 2023. — 352 с.
4. Фоксфорд. **Учебник. Информатика для школьников** [Электронный ресурс].
5. **Решу ОГЭ. Информатика.** (в том числе раздел «Задания, не входящие в экзамен этого года») [Электронный ресурс].

2 этап

Методика проведения испытания

Вступительное испытание 2-го этапа также проводится очно в Университетском Лицее №1511 или №1523 в соответствии со списками распределения (становятся известны накануне даты соответствующего вступительного испытания).

В течение 60 минут абитуриент решает задачи с проверкой написанного кода на наборе тестов в системе автоматизированной проверки решений Яндекс.Контест. Разрешённые языки для вступительного испытания — **Python, C/C++, Java**. Гарантируется, что задачи варианта возможно решить на полный балл на любом из этих языков. Для решения разных задач и для разных попыток сдачи конкретной задачи можно отдельно выбирать язык программирования из данного списка.

Количество баллов за каждую задачу будет описано в условии каждой задачи в разделе "Система оценивания". В случае полностью правильного решения задачи (положительного результата проверки на всём наборе тестов) выставляется максимальный балл за задачу и вердикт **ОК**; в случае частично верного решения (результат проверки положительный только на некоторых тестов из всего набора) выставляется пропорциональный балл по отношению к количеству пройденных тестов; в случае полностью неверного решения выставляется 0 баллов.

Во время написания экзамена нельзя пользоваться любыми справочными материалами, гаджетами, smartчасами, нейросетями, средствами аудио и видеосвязи, переговариваться с остальными

абитуриентами. В случае нарушения данных правил вы можете быть аннулированы и удалены с прохождения испытания.

Описание некоторых вердиктов тестирующей системы после отправки решения участника:

OK Тест пройден корректно;

WA Неверный ответ на тест — ответ на тест от программы участника не совпадает с верным ответом на тест;

TL Превышение времени работы программы — на конкретном тесте программа выполняется дольше указанного в задаче времени на выполнение;

ML Превышение используемой памяти в программе — на конкретном тесте программа занимает большую память, чем указано в ограничениях к памяти в задаче;

RE Ошибка исполнения программы — во время исполнения кода на данных из теста возникла программная ошибка;

SE Ошибка компиляции программы — во время компиляции программы произошла ошибка, на тестирование решение отправлено быть не может;

MI — Решение аннулировано в связи с нарушением правил проведения экзамена.

Максимальный балл за выполнение испытания 1-го этапа — 100 баллов, и вычисляется как сумма полученных баллов за все решённые в тестирующей системе задания.

Перечень разделов и тем для подготовки

Арифметические операции.

Арифметические выражения и их вычисление. Приоритет операторов, порядок вычислений. Операции сложения, вычитания, умножения, деления (в том числе целочисленного), остатка от деления. Типы данных в языках Python, C++, Java. Преобразование типов. Ошибки целочисленного деления и особенности работы оператора // в Python, оператора / и % в C++ и Java. Решение задач на вычисления, составление арифметических выражений, анализ значения переменных.

Условный оператор.

Понятие условия. Булевы выражения: сравнение чисел, логические операции, составные условия. Конструкции ветвления: `if`, `if-else`, `else-if` (Python: `elif`). Особенности синтаксиса условий в Python, C++, Java. Вложенные условия, каскад условий, корректность логики программы. Решение задач с разветвляющимся алгоритмом, производных от следующих базовых алгоритмов: выбор максимума, проверка свойств числа, ветвление по диапазонам.

Циклы с условием и счётчиком.

Назначение циклов. Циклы с известным числом повторений (`for`) и циклы с условием (`while`). Переменные-счётчики и управляющие переменные. Порядок изменения значений. Организация циклов в Python (`for ... in range`, `while`), C++ (`for`, `while`), Java (`for`, `while`). Бесконечные циклы и условия остановки. Решение задач, производных от следующих базовых алгоритмов: подсчёт суммы и количества элементов, поиск минимума/максимума, обработка цифр числа, перебор вариантов.

Одномерные массивы.

Представление массива в памяти. Индексация элементов: нумерация с нуля (Python, C, C++, Java). Обращение к элементам массива через индекс.

Создание и инициализация массивов: списки в Python, статические и динамические массивы в C и C++, массивы в Java. Правила задания длины массива, доступ к элементам, попытка выхода за границы массива как типовая ошибка.

Последовательный обход массива: цикл по индексам и цикл по элементам (Python). Получение значений элементов, их изменение, обработка каждого элемента массива в цикле. Типовые алгоритмы: подсчёт количества элементов, удовлетворяющих условию; вычисление суммы элементов; поиск максимального и минимального значений; поиск индекса нужного элемента; проверка наличия элемента.

Использование массивов для решения задач перебора: анализ числовых данных, обработка последовательностей, формирование новых массивов по заданному правилу. Типичные ошибки при работе с массивами: неправильная инициализация, неверные границы циклов, изменение массива во время перебора.

Двумерные массивы.

Понятие двумерного массива (матрицы, таблицы) как структуры данных, имеющей два индексных измерения: строку и столбец. Представление двумерного массива как массива массивов (Python, C, C++, Java). Индексация: порядок указания индексов (строка, затем столбец), от 0 или от 1 — в зависимости от языка.

Перебор двумерного массива: вложенные циклы. Прямой обход по строкам, обход по столбцам, обход по выбранным областям таблицы. Алгоритмы обработки таблиц: подсчёт количества элементов, удовлетворяющих условию; сумма элементов строки или столбца; поиск максимального/минимального элемента; подсчёт значений в прямоугольной области; сравнение элементов внутри строки или столбца.

Литература и интернет-источники для подготовки

1. **Яндекс.Лицей. Самостоятельные онлайн-курсы по программированию на Python** [Электронный ресурс].
2. **Образовательный центр «Сириус». Онлайн-платформа программирования для школьников** [Электронный ресурс].
3. **Решу ОГЭ. Информатика : задания по программированию для подготовки школьников** [Электронный ресурс].