

УТВЕРЖДАЮ

Директор АНОО

«Физтех-лицей» им. П.Л. Капицы

М.Г. Машкова

2022г.

ПРОГРАММА
промежуточной аттестации по информатике
2021–2022 учебный год

8 класс

(предпрофиль)

Практический минимум по алгоритмам

1. Выделение цифр из целого числа (с известным заранее и неизвестным заранее количеством цифр). Нахождение суммы, количества, максимальной, минимальной цифры в натуральном числе.
2. Нахождение суммы, количества, максимального и минимального элемента в последовательности числовых элементов (с заранее известным количеством элементов и с признаком конца).
3. Одновременный поиск двух максимальных элементов последовательности (с двумя возможными определениями второго максимума).
4. Поиск количества максимальных элементов в последовательности за однократный просмотр.
5. Поиск максимального элемента последовательности, обладающего определённым свойством (например, чётного).
6. Нахождение числа Фибоначчи по его номеру. Обратная задача.
7. Алгоритм Евклида нахождения НОД методом деления с остатком (идеальная реализация).
8. Проверка числа на простоту, эффективный алгоритм.
9. Факторизация (разложение числа на простые делители). Эффективный алгоритм.
10. Простой поразрядный перебор.
11. Обработка последовательности символов с признаком конца.
12. Алгоритмы обработки одномерных массивов: заполнение, вывод, переворот, сдвиг на 1 элемент и эффективный на k элементов, разбрасывание, сортировка (выбор максимума, пузырек, вставки), линейный и бинарный поиск.

Теоретические вопросы

1. Теория алгоритмов

- 1.1 Алгоритм. Свойства алгоритма. Способы записи алгоритма. Базовое понимание временной сложности алгоритма.
- 1.2 Понятие блок-схемы. Основные элементы блок-схем.
- 1.3 Основные алгоритмические конструкции: ветвление, цикл, линейный участок. Примеры задач на каждый вид.

2. Синтаксис. Базовые понятия в языке C++

- 2.1 Общие языковые понятия: алфавит, слово, синтаксис и семантика. Правила построения синтаксических диаграмм. Примеры.
- 2.2 Понятие переменной. Правила именования. Тип переменной в C++. Описание.
- 2.3 Понятие оператора (statement) и операции (operator). Арифметические операции. Операции сравнения. Логические операции. Тернарная условная операция. Операция последовательного применения команд. Операция присваивания. Ее синтаксис и принцип работы.
- 2.4 Базовое понимание категорий значения lvalue и rvalue.
- 2.5 Приоритет операций в C++.
- 2.6 Целые типы в C++. Логический тип данных bool.

2.7 Особенности вычисления операций. Явление переполнения. Тип результата. Преобразование типов при вычислениях. Операция явного приведения к типу.

2.8 Условный оператор `if`. Синтаксические диаграммы, принципы работы. Стратегии грамотного разбора случаев.

2.9 Операторы цикла `while` и `do while`. Синтаксические диаграммы, сопоставление принципов работы, примеры задач.

2.10 Оператор цикла `for`. Синтаксическая диаграмма, принцип работы, примеры задач. Реализация через `while`. Вложенные циклы, примеры задач.

2.11 Операторы `break` и `continue` применительно к циклам. Отличия в работе данных операторов в зависимости от вида оператора цикла.

2.12 Поточный ввод и вывод в C++. Концепция флагов, функций-членов `std::cout`, манипуляторов при форматном потоковом выводе.

2.13 Структуры данных `struct` в C. Синтаксис описания и использования. Использование `std::pair`, особенности.

2.14 Правила оформления программ: принципы форматирования, именования переменных, функций.

3. Числовые алгоритмы

3.1 Простые числа. Алгоритм проверки натурального числа на простоту методом перебора делителей (эффективный вариант).

3.2 Основная теорема арифметики (б/д). Алгоритм факторизации натурального числа. Обоснование алгоритма.

3.3 Алгоритм Евклида. Доказательство корректности (сходимость алгоритма к НОД).

3.4 Последовательность в математике. Линейная возвратная (рекуррентная) последовательность. Примеры. Линейный алгоритм вычисления n -го числа Фибоначчи при помощи трех переменных.

4. Функции в C++

4.1 Понятие функции в языках программирования. Структура функции в C++. Формальные и фактические параметры. Синтаксис описания и вызова функции в C++.

4.2 Прототипы функций (forward declaration). Зачем используются, общий синтаксис, примеры.

4.3 Оператор `return`. Особенности возврата результата. Функции типа `void`.

4.4 Механизмы передачи параметров в функции: по значению, по указателю, по ссылке. Особенности передачи массивов и контейнеров (`std::vector`, `std::string`) в функцию.

4.5 Параметры функции по умолчанию.

4.6 Понятие области видимости `Scope` в C++. Вложенные области видимости. Коллизия имен. Примеры.

4.7 Принцип локализации имен. Локальные и глобальные переменные. Спецификатор памяти `static` применительно к `Scope`-переменным, и переменным глобального блока.

4.8 Квалификатор типа `const`. Особенности применения с указательными типами.

5. Работа с символами и строками

5.1 Кодировочные таблицы. Устройство таблицы ASCII.

5.2 Символьный тип данных `char`. Работа с символами в C++. Специальные символы. Чтение до конца строки и до конца файла. Примеры задач.

5.3 Контейнер `std::string`. Основные методы. Обработка строковых данных.

6. Массивы. Указательная арифметика

6.1 Понятие одномерного массива. Синтаксис работы с массивами в C. Индексация.

6.2 Указатели в C. Описание указателей. Операции разыменования, взятия адреса и копирования. Приоритет операций `*` и `&`.

6.3 Адресная арифметика. Сравнение указателей.

6.4 Реализация многомерных массивов. Способы объявления, индексирование, передача в функцию. Объем занимаемой памяти. Способ хранения.

6.5 Двумерные массивы. Основные элементы: главная и побочная диагонали. Алгоритм поворота. Различные способы заполнения.

6.6 Указатель текущей позиции. Обработка массивов при помощи указателей. Замена операции индексирования в одномерном и двумерном массивах при помощи арифметики указателей.

6.7 Задача сортировки. Алгоритмы сортировки пузырьком и выбором максимума. Доказательство квадратичной сложности.

6.8 Алгоритм бинарного поиска. Доказательство корректности.

6.9 Бинарный поиск по ответу: общая идея, примеры задач. Экспоненциальный (галлопирующий) бинарный поиск.

6.10 Сортировка методом вставок. Доказательство квадратичной сложности. Оптимизация до бинарных вставок: идея, в чем выигрыш, итоговая сложность.

6.11 Сортировка подсчетом, если известно, что числа находятся в диапазоне

$$D = [min, max].$$

6.12 Динамические массивы. Контейнер `std::vector` и его основные методы.

7. Динамическое программирование

7.1 Метод динамического программирования при решении задач.

7.2 Одномерная динамика: вычисление последовательностей, количества вариантов, поиск оптимальных решений.

7.3 Задача LIS (о наибольшей возрастающей подпоследовательности) с восстановлением ответа.

7.4 Двумерная динамика. Основные задачи. Задача LCS (о наибольшей общей подпоследовательности).

7.5 Расстояние Левенштейна.

7.6 Задача о классическом рюкзаке. Задача о рюкзаке с повторениями.

8. Системы счисления

8.1 Системы счисления. Основные определения: базис, основание, алфавит. Традиционные (ричные) системы счисления. Свёрнутая и развёрнутая форма записи числа.

8.2 Алгоритм перевода с использованием арифметики целевой системы счисления (для рациональных чисел) с обоснованием.

8.3 Алгоритм перевода периодических дробей с использованием арифметики целевой системы счисления с обоснованием.

8.4 Алгоритмы перевода целых чисел с использованием арифметики исходной системы счисления: делением и выделением максимальной степени с обоснованием. Их сравнительный анализ.

8.5 Алгоритм перевода рациональных чисел с использованием арифметики исходной системы счисления с обоснованием.

8.6 Быстрый алгоритм перевода для систем счисления, основания которых связаны соотношением $p^m = q$ с обоснованием.

Тематика задач

- Счет в различных системах счисления, базовые задачи на понимание чисел.
- Понимание динамики (математика), задачи на расстояние Левенштейна, на выравнивание строк.
- Линейные рекуррентные последовательности (вычисление, обработка).
- Вычисление сложных C-выражений.
- Написание кода: структуры, компараторы.