

**Московский Государственный Университет
имени М.В. Ломоносова**

Механико-математический факультет

Учебно-методический комплекс дисциплины

Основы ЭВМ и программирование

Специальность: механика 010901

Квалификация выпускника: механик

Форма обучения: очная дневная

Автор: доцент кафедры вычислительной математики механико-математического факультета МГУ, кандидат физ-мат наук
Пронкин Юрий Николаевич

**Москва
2013**

1. Название дисциплины

Основы ЭВМ и программирование

2. Цель и задачи дисциплины

2.1. Цели дисциплины

- Приобретение начального опыта использования вычислительных систем для решения научно-практических задач.
- Приобретение знаний и навыков, необходимых для последующей работы в коллективах разработчиков программного обеспечения вычислительных систем.
- Приобретение базовых знаний в области основ вычислительной техники, теории алгоритмов и методов вычислений, необходимых для успешного дальнейшего освоения специальных дисциплин.

2.2. Задачи дисциплины

- Формирование навыков работы на вычислительных системах под управлением операционных систем Windows и Linux.
- Приобретение опыта использования вычислительных систем в офисной практике и для решения прикладных задач механики и математики.
- Освоение языков программирования и приобретение базовых навыков разработки прикладного и системного программного обеспечения.
- Приобретение базовых знаний в области теории алгоритмов и методов вычислительной математики.
- Освоение современных технологий разработки и тестирования программного обеспечения.
- Приобретение базовых знаний в области архитектуры вычислительных систем, сетевого оборудования и периферийных устройств.
- Приобретение базовых знаний в области архитектуры и технологии разработки операционных систем.

2.3. По результатам освоения дисциплины обучающийся должен:

Знать

- Основы аппаратно-программной архитектуры современных вычислительных систем.
- Принципы организации операционных систем и основы системного программирования.
- Современные языки программирования.
- Модели и методы представления данных в вычислительных системах
- Базовые алгоритмы обработки данных.
- Основы использования и программирования компьютерных сетей
- Основы технологии разработки программного обеспечения и программных проектов .

Уметь

- Реализовывать программы решения научно-практических задач на современных языках программирования.
- Использовать вычислительные системы для проведения научных и прикладных вычислений.
- Применять вычислительные системы в повседневной и офисной практике.

3. Место дисциплины в структуре ООП

3.1. Место дисциплины в учебном плане

- Семинары: 1 – 2 курс, 1 – 4 семестры
- Лекции: 1 – 2 курс, 2 – 3 семестры

3.2. Перечень дисциплин, которые должны быть освоены для начала освоения и параллельно данной дисциплине

- Элементарная математика в объеме программы средней школы и элементы высшей математики.
- Основы математического анализа.
- Основы линейной алгебры.

3.3. Общая трудоемкость дисциплины

- 328 академических часов

3.4. Формы промежуточной аттестации

- Зачеты (семестры 1, 2 и 4).
- Экзамен (семестр 3).

3.5. Связь с другими дисциплинами

Данная дисциплина дает базовые знания в области алгоритмических языков, программирования, организации вычислительных систем, общей теории алгоритмов и методов вычислений создавая тем самым фундамент для углубленного изучения специальных разделов теории алгоритмов, технологии баз данных, методов вычислительной математики и ряда других смежных дисциплин.

4. Форма проведения занятий

- Аудиторная работа, лекции – 72 часа
- Аудиторная работа, семинары – 144 часа
- Самостоятельная работа – 112 часов

5. Распределение трудоемкости по разделам, темам и формам проведения занятий

Изучение настоящей дисциплины предполагает:

- Проведение семинарских и практических занятий в группах (1 – 4 учебные семестры)
- Чтение лекционного курса для всех групп потока (2 – 3 учебные семестры)

- Проведение промежуточного и итогового тестирования по результатам работы в каждом семестре

Общий объем учебной работы по семестрам (часы)

Вид учебной работы	Семестр 1	Семестр 2	Семестр 3	Семестр 4
Лекционный курс	0	28	44	0
Семинарские занятия	44	44	28	28
Самостоятельная работа	28	28	28	28
Всего часов	72	100	100	56

Форма отчетности по семестрам

Вид учебной работы	Семестр 1	Семестр 2	Семестр 3	Семестр 4
Лекционный курс	Нет	Нет	Экзамен	Нет
Семинарские и практические занятия	Зачет	Зачет	Экзамен	Зачет

Трудоемкость по темам и формам проведения занятий (часы)

№ п/п	Наименование разделов и тем дисциплины	Трудоемкость (в ак. часах) по формам занятий			Формы контроля
		Аудиторная работа с разбивкой по формам и видам		Самостоятельная работа	
		Лекции	Семинары		
1	Тема 1.1. Введение в программирование на языке Си	0	12	12	Домашнее задание
2	Тема 1.2. Обработка последовательностей и однопроходные алгоритмы	0	16	16	Контрольная работа
3	Тема 1.3. Обработка одномерных массивов	0	16	16	Зачет
4	Тема 2.1. Обработка многомерных массивов	0	12	12	Контрольная работа
5	Тема 2.2. Решение систем линейных алгебраических уравнений прямыми и итерационными методами	0	12	12	Контрольная работа
6	Тема 2.3. Обработка символьных строк и текстов	0	10	10	Контрольная работа
7	Темы 2.4. Представление и обработка структур данных в языке Си	0	10	10	Зачет
8	Тема 3.1. Моделирование динамических структур данных с последовательным доступом	0	8	8	Контрольная работа
9	Тема 3.2. Моделирование ссылочных динамических структур данных	0	14	14	Контрольная работа
10	Тема 3.3. Алгоритмы с использованием двоичных деревьев	0	6	6	Контрольная работа

11	Тема 4.1. Введение в методологию разработки программных систем и архитектуру аппаратно-программного обеспечения	2	0	0	
12	Темы 4.2. Общая архитектура микропроцессорных вычислительных систем	8	0	0	
13	Темы 4.3. Многоуровневая организация памяти	4	0	0	
14	Темы 4.4. Организация системной шины	2	0	0	
15	Тема 4.5. Машинные способы представления информации	4	0	0	
16	Тема 4.6. Введение в архитектуру микропроцессоров	4	0	0	
17	Темы 4.7. Организация работы с подпрограммами	2	0	0	
18	Тема 4.8. Прерывания и программы обработки прерываний	2	0	0	
19	Тема 4.9. Обработка внешних аппаратных прерываний	2	0	0	
20	Тема 4.10. Работа микропроцессоров Intel x86 в режиме защищенного адреса	6	0	0	Коллоквиум
21	Тема 5.1. Введение в операционные системы	4	0	0	
22	Тема 5.2. Управление процессами	2	0	0	
23	Тема 5.3. Основные задачи системного программирования	4	0	0	
24	Тема 6.1. Введение в архитектуру языков программирования	2	0	0	
25	Тема 6.2. Управление данными в языках программирования	4	0	0	
26	Тема 6.3. Управление последовательностью действий в языках программирования	2	0	0	
27	Тема 7.1. Введение в методы структурирования и агрегирования данных	2	0	0	
28	Тема 7.2. Моделирование динамических структур данных с последовательным доступом	2	0	0	
29	Тема 7.3. Ссылочные реализации динамических структур данных	4	0	0	
30	Тема 7.4. Представление и обработка множеств	2	0	0	
31	Тема 7.5. Базовые алгоритмы сортировки и поиска	4	0	0	
32	Тема 7.6. Элементарное введение в файловые	2	0	0	

	системы				
33	Тема 8.1. Основы построение компьютерных сетей	2	0	0	Экзамен
34	Тема 9.1. Работа над индивидуальными проектами в области представления и обработки реляционных баз данных	0	28	28	Зачет
35	Итого:	72	144	112	

6. Содержание дисциплины

Раздел 1. Обработка одномерных структур данных в программах на языке Си (семинары - 44 часа, самостоятельная работа - 44 часа)

Данный раздел является первым вводным разделом в программирование на языке Си. В нем студенты изучают основы языка Си, приобретают навыки разработки простейших программ и знакомятся базовыми с алгоритмами обработки последовательностей и одномерных массивов.

Тема 1.1. Введение в программирование на языке Си (семинары - 12 часов, самостоятельная работа - 12 часов)

- Общая структура Си-программы.
- Компиляция и сборка программных модулей.
- Алфавит языка, именование объектов программы, ключевые слова.
- Базовые и производные типы данных.
- Константы.
- Операции.
- Операторы.
- Работа с функциями и механизмы передачи параметров.
- Примеры программ.

Тема 1.2. Обработка последовательностей и однопроходные алгоритмы (семинары - 16 часов, самостоятельная работа - 16 часов)

- Понятие последовательности и однопроходного алгоритма.
- Индуктивные функции и индуктивные расширения.
- Примеры реализации алгоритмов и программ.
- Самостоятельная работа над задачами.

Тема 1.3. Обработка одномерных массивов (семинары - 16 часов, самостоятельная работа - 16 часов)

- Статическое и динамическое представление одномерных массивов в языке Си.
- Базовые алгоритмы сортировки (методы «пузырька» и QuickSort).
- Примеры реализации алгоритмов и программ.
- Самостоятельная работа над задачами.

Раздел 2. Обработка многомерных структур данных в программах на языке Си (семинары - 44 часа, самостоятельная работа - 44 часа)

В данном разделе студенты осваивают более сложные конструкции языка Си такие, как аппарат указателей, побитовые операции и работа со структурами. Необходимые практические навыки приобретаются в

результате работы над задачами на обработку многомерных числовых массивов, численное решение систем линейных алгебраических уравнений, обработку символьных строк и текстовых массивов и обработку структурированных неоднородных данных.

Тема 2.1. Обработка многомерных массивов (семинары – 12 часов, самостоятельная работа – 12 часов)

- Статическое и динамическое представление многомерных массивов в языке Си.
- Базовые алгоритмы работы с числовыми матрицами.
- Примеры реализации алгоритмов и программ.
- Самостоятельная работа над задачами.

Тема 2.2. Решение систем линейных алгебраических уравнений прямыми и итерационными методами (семинары – 12 часов, самостоятельная работа – 12 часов)

- Введение в прямые и итерационные методы решения систем линейных алгебраических уравнений (СЛАУ).
- Прямой метод Гаусса-Жордана.
- Итерационные методы простой итерации и скорейшего спуска, сравнительное исследование скорости сходимости для различных матриц системы.
- Самостоятельная работа над задачами.

Тема 2.3. Обработка символьных строк и текстов (семинары – 10 часов, самостоятельная работа – 10 часов)

- Представление символьных строк и текстовых массивов в языке Си.
- Функции стандартной библиотеки для работы со строками.
- Характерные задачи и алгоритмы обработки строк и текстовых массивов.
- Примеры реализации алгоритмов и программ.
- Самостоятельная работа над задачами.

Темы 2.4. Представление и обработка структур данных в языке Си (семинары – 10 часов, самостоятельная работа 10 часов)

- Структуры и объединения в языке Си, операции декомпозиции.
- Указатели на структурированные объекты и динамическое выделение памяти.
- Решение задач на использование структур и объединений.

Раздел 3. Моделирование динамических структур данных в программах на языке Си (семинары - 28 часов, самостоятельная работа - 28 часов)

Данный раздел знакомит студентов с непрерывными и ссылочными динамическими структурами данных и методами их моделирования средствами языка Си. Программные реализации выполняются с учетом требований современных технологий программирования, предполагают написание автоматизированных тестов и использование динамических структур данных при реализации различных прикладных задач.

Тема 3.1. Моделирование динамических структур данных с последовательным доступом (семинары – 8 часов, самостоятельная работа – 8 часов)

- Структура данных «Стек».
 - Дисциплина добавления и исключения элементов.
 - Программная реализация.
 - Реализация автоматизированных тестов.
 - Реализация прикладных алгоритмов, использующих структуру данных «Стек».
- Структура данных «Очередь».
 - Дисциплина добавления и исключения элементов.
 - Программная реализация.
 - Реализация автоматизированных тестов.
 - Реализация прикладных алгоритмов, использующих структуру данных «Очередь».

Тема 3.2. Моделирование ссылочных динамических структур данных (семинары – 14 часов, самостоятельная работа – 14 часов)

- Структура данных «Двунаправленный список».
 - Дисциплина добавления и исключения элементов.
 - Программная реализация.
 - Реализация автоматизированных тестов.
 - Реализация прикладных алгоритмов, использующих структуру данных «Двунаправленный список».
- Структура данных «Дерево».
 - Дисциплина добавления и исключения элементов.
 - Программная реализация.
 - Реализация автоматизированных тестов.
 - Реализация прикладных алгоритмов, использующих структуру данных «Дерево».

Тема 3.3. Алгоритмы с использованием двоичных деревьев (семинары – 6 часов, самостоятельная работа – 6 часов)

- Программная реализация алгоритмов обхода двоичного дерева.
- Программная реализация деревьев поиска.
- Программная реализация алгоритмов сортировки, использующих структуру бинарного дерева (метод HeapSort).

Раздел 4. Основы аппаратно-программной организации вычислительных систем (лекции - 36 часов)

В данном разделе рассматриваются наиболее важные элементы архитектуры вычислительных систем, адресация основной памяти компьютера, модели ассоциативной кэш-памяти, вопросы архитектуры микропроцессоров, машинные способы представления информации, организация работы с подпрограммами, управление многозадачностью.

Тема 4.1. Введение в методологию разработки программных систем и архитектуру аппаратно-программного обеспечения (лекция - 2 часа)

- Общий подход к решению задач на вычислительных системах.
- Общая архитектура аппаратно-программного обеспечения вычислительных систем.

Тема 4.2. Общая архитектура микропроцессорных вычислительных систем (лекции - 8 часов)

- Функциональная схема компьютера.
- Предназначение и функционирование основных элементов аппаратуры.
- Типы и характеристики микропроцессоров (разрядность, частота синхронизации, организация системы команд).
- Конвейерная обработка команд.
- Классификация и способы разрешения конфликтов при работе конвейера. Сравнение CISC и RISC архитектур.
- Организация основной памяти компьютера, понятие адресного пространства, реальные и виртуальные адреса.
- Прямой доступ к основной памяти.

Тема 4.3. Многоуровневая организация памяти (лекции - 4 часа)

- Концепции и предпосылки многоуровневой организации памяти.
- Модели ассоциативной кэш-памяти и вопросы оптимизации.
- Размещение, поиск и замещение блоков.
- Стратегии записи.
- Многоуровневая кэш-память.

Тема 4.4. Организация системной шины (лекция - 2 часа)

- Взаимодействие основных аппаратных модулей.
- Управление доступом к основной памяти и подсистеме ввода/вывода.
- Классификация и адресация портов ввода/вывода.

Тема 4.5. Машинные способы представления информации (лекции - 4 часа)

- Представление числовых и символьных данных в вычислительных системах.
- Машинная арифметика.
- Влияние погрешности представления вещественных чисел на точность вычислений.

Тема 4.6. Введение в архитектуру микропроцессоров (лекции - 4 часа)

- Базовые элементы архитектуры микропроцессора.
- Управление потоком команд.
- Базирование и индексирование памяти.
- Управление стеком.
- Битовые флаги состояния и управления.
- Примеры конкретных архитектур.

Тема 4.7. Организация работы с подпрограммами (лекции - 2 часа)

- Концепция подпрограмм и функций, схема взаимных вызовов.
- Рекурсивные алгоритмы и рекурсивные вызовы.
- Семантика команд вызова и возврата.
- Вызовы с параметрами, механизмы передачи параметров.

Тема 4.8. Прерывания и программы обработки прерываний (лекция - 2 часа)

- Классификация прерываний.
- Организация таблицы векторов прерываний.

- Внутренние прерывания микропроцессора и зарезервированные вектора прерываний.
- Последовательность прерывания.
- Примеры реализации и использования подсистемы прерываний.

Темы 4.9. Обработка внешних аппаратных прерываний (лекция - 2 часа)

- Назначение и работа контроллера прерываний.
- Приоритеты внешних устройств.
- Последовательность аппаратного прерывания.
- Примеры реализации.

Тема 4.10. Работа микропроцессоров Intel x86 в режиме защищенного адреса (лекции - 6 часов)

- Общая схема формирования исполнительного адреса.
- Кольца защиты.
- Адресное пространство задачи.
- Организация дескрипторных таблиц.
- Работа с подпрограммами в режиме защищенного адреса, межкольцевые вызовы подпрограмм, ограничение прав доступа и отражение атаки «Троянского коня».
- Многозадачность, управление задачами, сегмент состояния задачи, вложение задач.

Раздел 5. Архитектура операционных систем и основные задачи системного программирования (лекции – 10 часов)

В данном разделе рассматриваются общие вопросы организации операционных систем, приводится классификация операционных систем, обсуждается понятие процесса и диаграммы его состояния, рассматриваются классические задачи системного программирования.\

Тема 5.1. Введение в операционные системы (лекции - 4 часа)

- Вычислительные ресурсы, классификация ресурсов, виртуализация ресурсов, задача управления ресурсами.
- Понятие операционной системы, классификация операционных систем.
- Иерархия подсистем, системные вызовы, примеры реализации и использования.

Тема 5.2. Управление процессами (лекция - 2 часа)

- Понятие процесса.
- Диаграммы состояний процесса и их обсуждение.
- Параллельное выполнение программ в различных операционных средах, многопоточные операционные системы.

Темы 5.3. Основные задачи системного программирования (лекции - 4 часа)

- Введение в задачи системного программирования.
- Задача управления реальной и виртуальной памятью.
- Взаимодействие процессов, асинхронно выполняющиеся процессы.
- Проблема критической секции и методы ее решения.

- Проблема тупика и методы ее решения.

Раздел 6. Архитектура языков программирования (лекции - 8 часов)

В данном разделе студенты знакомятся с вопросами внутренней архитектуры и реализации языков программирования, методами управления памятью, данными и последовательностью действий, механизмами вызова подпрограмм и передачи параметров.

Тема 6.1. Введение в архитектуру языков программирования (лекция - 2 часа)

- История развития и иерархия языков программирования.
- Модульная архитектура языков программирования (на примере языка Фортран).
- Блочная архитектура языков программирования (на примере языка Алгол).
- Модульно-блочная структура Си-программы.

Тема 6.2. Управление данными в языках программирования (лекции - 4 часа)

- Модели данных языков программирования.
- Статическая и динамическая память.
- Глобальные, локальные и нелокальные среды.
- Базовые типы данных в языке Си и методы конструирования производных типов данных.
- Понятие классов памяти, управление классами памяти в языке Си.

Тема 6.3. Управление последовательностью действий в языках программирования (лекция - 2 часа)

- Операторы управления последовательностью действий.
- Вызовы функций и механизмы передачи параметров.

Раздел 7. Структуры данных и алгоритмы (лекции - 16 часов)

В данном разделе с общетеоретических и практических позиций рассматриваются вопросы абстрагирования и структурирования данных, методы моделирования динамических структур данных с последовательным и ссылочным доступом, способы представления множеств и эффективные алгоритмы поиска, а также базовые алгоритмы сортировки.

Тема 7.1. Введение в методы структурирования и агрегирования данных (лекция - 2 часа)

- Классификация структур данных.
- Общий подход к абстрагированию и структурированию данных.
- Понятие объекта и объектно-ориентированные технологии.

Тема 7.2. Моделирование динамических структур данных с последовательным доступом (лекция - 2 часа)

- Последовательности и однопроходные алгоритмы, индуктивные функции и индуктивные расширения.
- Непрерывные реализации динамических структур данных на примере стека, дека и очереди.

Тема 7.3. Ссылочные реализации динамических структур данных (лекции - 4 часа)

- Концепция ссылочных реализаций.
- Одно- и двунаправленные списки.
- Представление деревьев и графов.
- Алгоритмы обхода двоичного дерева.
- Деревья поиска.
- Примеры использования деревьев в вычислительных алгоритмах.

Тема 7.4. Представление и обработка множеств (лекция - 2 часа)

- Понятие множества.
- Последовательный и двоичный поиск.
- Битовая реализация множества.
- Оптимизация поиска и задача хеширования множеств. Примеры построения хеш-функций.

Тема 7.5. Базовые алгоритмы сортировки и поиска (лекции - 4 часа)

- Априорная оценка вычислительной сложности алгоритма.
- Сортировка методами «пузырька», QuickSort, HeapSort и Шелла.
- Алгоритмы поиска подстрок.
- Примеры реализации алгоритмов сортировки и поиска.

Тема 7.6. Элементарное введение в файловые системы (лекция - 2 часа)

- Представление файлов и каталогов.
- Обзор файловых систем FAT, NTFS и EXT

Раздел 8. Введение в организацию компьютерных сетей и сетевые технологии (лекция - 2 часа)

В данном разделе дается вводное представление о современных сетевых технологиях, протоколах и методах сетевого доступа. Полученные знания могут служить базисом для дальнейшего изучения дисциплины в рамках специальных курсов.

Тема 8.1. Основы построения компьютерных сетей (лекция - 2 часа)

- Классификация компьютерных сетей.
- Пространство Интернет, понятие Интернет-протокола
- Адресация в IP-сетях.

Раздел 9. Разработка индивидуальных проектов (семинары - 28 часов, самостоятельная работа - 28 часов)

В данном разделе курса слушатели впервые приступают к самостоятельной работе над относительно крупными программными проектами в области технологий баз данных. Важным элементом учебного процесса является то, что студенты выступают в качестве авторов тематики конкретных проектов и их функционального наполнения, оставляя за преподавателем, главным образом, консультативную роль.

Тема 9.1. Работа над индивидуальными проектами в области представления и обработки реляционных баз данных (семинары - 28 часов, самостоятельная работа - 28 часов)

- Постановка задачи на разработку индивидуального проекта
- Обсуждение тематики проектов и основной функциональности
- Проектирование и обсуждение модели представления данных
- Проектирование и обсуждение модели организации программного кода
- Проектирование и обсуждение модели доступа к базе данных по технологии «клиент-сервер»
- Программная реализация проекта
- Документирование проекта
- Защита проекта

7. Используемые образовательные, научно-исследовательские и научно-производственные технологии

- **Образовательные технологии:** интерактивные лекции и семинары, решение типовых задач, выполнение творческого задания, дискуссии по теме занятий, активное обсуждение и оценка работы студентов в группе, самостоятельная работа.
- **Научно-исследовательские технологии:** изучение литературы, а также научных и научно-популярных статей, блогов и лекций ведущих отечественных и зарубежных специалистов, представленным в Интернете.

8. Учебно-методическое обеспечение самостоятельной работы студентов, оценочные средства контроля успеваемости и промежуточной аттестации

8.1 Учебно-методические рекомендации для обеспечения самостоятельной работы

В первых двух учебных семестрах следует сконцентрироваться на хорошем владении основными инструментами языка Си, освоить базовые алгоритмы обработки числовых последовательностей и массивов, обратить внимание на хороший стиль программирования. Надежность разрабатываемых программ и их эффективность на этом этапе могут и не занимать первостепенного места. В третьем семестре следует обращать существенное внимание на эффективность реализации алгоритмов и представления структур данных. Важное место здесь приобретает правильность и надежность функционирования программного кода, достигаемая тщательным ручным тестированием и написанием автоматизированных тестов. В четвертом семестре работа над относительно крупными программными проектами ведется студентами по большей степени самостоятельно, оставляя за преподавателем лишь роль постановщика задачи в общем виде и консультанта в ходе выполнения проекта.

8.2 Примерный список заданий для проведения аттестаций

8.2.1. Примеры заданий по теме 1.2 на обработку последовательностей и однопроходные алгоритмы

1. Подсчитать количество элементов последовательности, большие предыдущего
2. Определить номер последнего элемента числовой последовательности, равного заданному числу X .
3. Найти общее количество элементов в постоянных участках целочисленной последовательности.
4. Определить номер последнего элемента числовой последовательности, равного минимуму всех элементов этой последовательности.
5. Найти количество возрастающих участков числовой последовательности.
6. Найти величину максимального отклонения элементов последовательности от их среднего арифметического.
7. Найти сумму четных элементов последовательности во всех ее возрастающих участках.
8. Найти среднее арифметическое локальных экстремумов числовой последовательности.
9. Найти количество локальных максимумов/минимумов числовой последовательности, лежащих левее/правее глобального минимума/максимума.

8.2.2. Примеры заданий по теме 1.3 на обработку одномерных массивов

1. Заменить каждый элемент массива на полусумму соседних с ним элементов (кроме первого и последнего).
2. Сгруппировать положительные элементы массива в его начале, а отрицательные в конце с сохранением их исходного порядка.
3. Двоичным поиском определить позицию неубывающего массива, в которую в этот массив можно вставить заданное число X .
4. Даны два неубывающих массива. Построить третий неубывающий массив, являющийся объединением первых двух (элементы в массивах могут повторяться).
5. Неотрицательные элементы массива расположить в его начале в порядке возрастания, а отрицательные элементы в конце в порядке убывания.
6. Назовем массив целых чисел плотным, если множество значений его элементов полностью заполняет некоторый отрезок $[a, b]$. Определить, является ли заданный массив плотным.
7. Выполнить сортировку элементов числового массива по возрастанию методом QuickSort.

8.2.3. Примеры заданий по теме 2.1 на обработку многомерных массивов

1. Матрицу вещественных чисел размерности $M \times N$ загрузить из файла, в котором первые два числовых значения задают размерность матрицы. Выполнить над загруженной матрицей следующее преобразование: найти строку, содержащую минимальный элемент матрицы, и удалить из матрицы эту строку.

2. Матрицу вещественных чисел размерности $M \times N$ загрузить из файла, в котором первые два числовых значения задают размерность матрицы. Выполнить над загруженной матрицей следующее преобразование: найти столбец с самым маленьким диагональным элементом и заменить в этом столбце каждый элемент на сумму остальных элементов строки, в которой он стоит.
3. Матрицу вещественных чисел размерности $M \times N$ загрузить из файла, в котором первые два числовых значения задают размерность матрицы. Выполнить над загруженной матрицей следующее преобразование: найти столбец, содержащий максимальный элемент матрицы, и заменить в этом столбце каждый элемент на максимальное значение из строки, в которой он стоит.
4. Матрицу вещественных чисел размерности $M \times N$ загрузить из файла, в котором первые два числовых значения задают размерность матрицы. Выполнить над загруженной матрицей следующее преобразование: найти столбец, содержащий минимальный элемент матрицы, и вычесть этот столбец из всех остальных столбцов матрицы, оставив сам этот столбец без изменений.
5. Матрицу вещественных чисел размерности $M \times N$ загрузить из файла, в котором первые два числовых значения задают размерность матрицы. Выполнить над загруженной матрицей следующее преобразование: найти строку с максимальной суммой элементов и заменить в этой строке каждый элемент на сумму остальных элементов столбца, к которому он стоит.

8.2.4. Примеры заданий по теме 2.3 на обработку символьных строк и текстов

1. Во входном файле записан некоторый текст, общая длина которого не известна. Требуется поставить в начале каждой строки (в том числе и пустой) ее порядковый номер, отделяя его от содержимого исходной строки одним дополнительным пробелом. Преобразованный текст записать в выходной файл.
2. Во входном файле записан некоторый текст, общая длина которого не известна. Требуется заключить каждое слово текста в квадратные скобки, считая словом всякую группу идущих подряд символов, не включенных в число разделителей. Преобразованный текст записать в выходной файл.
3. Во входном файле записан некоторый текст, общая длина которого не известна. В конце некоторых строк текста слова разорваны для переноса с помощью символа '-'. Требуется восстановить разорванные слова, перенеся их целиком на следующую строку. Преобразованный текст записать в выходной файл.

8.2.4. Примеры основных вопросов и задач экзаменационных билетов

Билет 1

1. Общая архитектура микропроцессорных вычислительных систем. Функциональная схема компьютера. Типы и характеристики микропроцессоров (разрядность, частота синхронизации, организация системы команд). Конвейерная обработка команд. Классификация и

- способы разрешения конфликтов при работе конвейера. Сравнение CISC и RISC архитектур.
2. Написать программу загрузки из текстового файла последовательности целых чисел, игнорирующую все элементы исходной последовательности, равные среднему арифметическому своих соседей. Память выделить динамически на максимальное количество элементов последовательности. Напечатать загруженные элементы в порядке их возрастания.

Билет 2

1. Типы и характеристики запоминающих устройств. Организация основной памяти компьютера. Понятие адресного пространства. Реальные и виртуальные адреса. Прямой доступ к основной памяти.
2. Написать программу, отыскивающую во входном файле целых чисел такие, у которых старший байт их внутреннего машинного представления является «зеркальным отражением» младшего байта (например, 01010100.bbbbbbbb.bbbbbbbb.00101010). Память выделить динамически на максимальное количество элементов последовательности. Напечатать найденные числа в порядке возрастания их абсолютной величины.

Билет 3

1. Многоуровневая организация памяти. Модели ассоциативной кэш-памяти. Размещение, поиск и замещение блоков. Стратегии записи. Многоуровневая кэш-память.
2. Написать программу сортировки последовательности целых чисел произвольным методом, принимая за ключ сортировки сумму всех цифр десятичной записи числа. Последовательность загрузить из файла, память для хранения выделить динамически. Напечатать отсортированную последовательность.

Билет 4

1. Организация системной шины персональных компьютеров. Взаимодействие основных аппаратных модулей. Управление доступом к основной памяти и подсистеме ввода/вывода. Классификация и адресация портов ввода/вывода.
2. Написать программу, загружающую из файла последовательность символьных строк ограниченной длины и располагающую их в элементах динамически создаваемого двунаправленного связного списка. Напечатать загруженные строки в лексикографическом порядке. Для сравнения строк воспользоваться функцией strcmp().

Билет 5

1. Представление данных в ЭВМ. Машинная арифметика. Погрешность представления чисел с плавающей точкой и ее влияние на точность вычислений.
2. Числовую матрицу с заданным количеством колонок и заранее неизвестным количеством строк загрузить из файла, выделяя память динамически под каждую ее строку. Написать функцию, печатающую элемент матрицы по его линейному индексу в предположении, что нумерация элементов матрицы ведется по колонкам, начиная с нуля.

Билет 6

1. Базовые элементы архитектуры микропроцессоров. Управление потоком команд. Базирование и индексирование памяти. Управление стеком. Битовые флаги состояния и управления. Организация памяти и формирование исполнительного адреса на примере реального режима микропроцессоров Intel x86.
2. Выполнить программную реализацию объекта «Стек двухкомпонентных векторов». Максимальная глубина стека задается динамически при инициализации объекта.

Билет 7

1. Организация работы с подпрограммами. Рекурсивные вызовы. Семантика команд вызова и возврата на примере режима реального адреса микропроцессоров Intel x86. Вызовы с параметрами. Механизмы передачи параметров.
2. Написать программу подсчета количества слов в произвольном текстовом файле, принимая за разделители слов любой из символов введенной с клавиатуры цепочки. Определить количество символов в самом длинном слове. Результаты напечатать.

Билет 8

1. Прерывания и программы обработки прерываний. Классификация прерываний. Зарезервированные вектора прерываний режима реального адреса микропроцессоров Intel x86. Последовательность прерывания.
2. Выполнить программную реализацию объекта «Очередь строк фиксированной длины» на базе кольцевого вектора. Максимальная длина очереди задается динамически при инициализации объекта.

Билет 9

1. Обработка внешних аппаратных прерываний. Назначение и работа контроллера прерываний. Двухуровневая схема контроллеров прерываний. Последовательность аппаратного прерывания.
2. Написать программу сортировки последовательности целых чисел методом QuickSort, принимая за ключ сортировки количество единиц в машинном представлении каждого числа. Последовательность загрузить из файла, память для хранения выделить динамически. Напечатать отсортированную последовательность и количество выполненных операций сравнения ключей.

Билет 10

1. Работа микропроцессоров Intel x86 в защищенном режиме. Общая схема формирования исполнительного адреса. Адресное пространство задачи. Кольца защиты. Организация дескрипторных таблиц. Типы и структура дескрипторов.
2. Написать программу, загружающую из файла последовательность символьных строк ограниченной длины с динамическим выделением памяти для хранения символов каждой строки. Отсортировать строки в порядке убывания количества слов в каждой строке.

Билет 11

1. Работа с подпрограммами в режиме защищенного адреса микропроцессоров Intel x86. Межкольцевые вызовы подпрограмм. Ограничение прав доступа и отражение «атаки Троянского коня». Управление задачами. Сегмент состояния задачи. Вложенные задачи.
2. Написать программу сортировки последовательности вещественных чисел методом HeapSort. Последовательность загрузить из файла, память для хранения выделить динамически. Напечатать отсортированную последовательность и количество выполненных операций сравнения элементов.

Билет 12

1. Задача управления ресурсами ЭВМ. Операционные системы и их классификация. Иерархия подсистем. Системные вызовы. Примеры использования системных функций.
2. Написать программу, загружающую из текстового файла массив записей, каждая из которых содержит одно строковое и одно числовое поле. Записи расположить в элементах динамически создаваемого связного списка. Напечатать загруженные записи в порядке убывания значений числового поля.

Билет 13

1. Понятие процесса. Диаграммы состояний процесса. Параллельное выполнение программ в различных операционных средах. Многопоточные операционные системы.
2. Написать программу загрузки из текстового файла последовательности символьных строк, игнорирующую все строки, содержащие хотя бы одну цепочку символов из заданного с клавиатуры набора цепочек. Память для хранения символов каждой строки выделить динамически. Напечатать загруженные строки в порядке возрастания количества символов в каждой строке.

Билет 14

1. Задача управления памятью. Связное и несвязное распределение физической памяти. Выделение памяти разделами фиксированного и переменного размера. Оверлейные структуры. Системы со свопингом. Виртуальная память. Сегментная и страничная организация виртуальной памяти. Трансляция адресов. Стратегии замещения разделов. Сегментно-страничная модель памяти микропроцессоров Intel x86.
2. Написать программу загрузки из текстового файла последовательности целых чисел, игнорирующую элементы исходной последовательности, равные сумме всех предыдущих загруженных элементов. Память выделить динамически на максимальное количество элементов последовательности. Напечатать загруженные элементы в порядке убывания их абсолютной величины.

Билет 15

1. Взаимодействие процессов. Асинхронно выполняющиеся процессы. Проблема критической секции и семафорные примитивы. Проблема тупика и методы ее решения. Алгоритм банкира.

2. Числовую последовательность неопределенной длины загрузить из текстового файла, выделяя память динамически по мере необходимости блоками фиксированного размера. Напечатать загруженную последовательность в порядке возрастания элементов.

Билет 16

1. Модели данных языков программирования. Статическая и динамическая память. Модульность и блочность. Глобальные и локальные среды. Управление данными в Си программах. Классы памяти. Методы структурирования данных. Вызовы функций и механизмы передачи параметров.
2. Написать программу подсчета общего количества и процентного отношения появления каждого символа в текстовом файле. Результат представить в виде таблицы в порядке возрастания частот появления различных символов.

Билет 17

1. Общий подход к абстрагированию и структурированию данных. Понятие объекта и объектно-ориентированные технологии. Последовательности и однопроходные алгоритмы. Непрерывные реализации динамических структур данных на примере стека и очереди.
2. Написать программу, загружающую из текстового файла последовательность символьных строк ограниченной длины и располагающую их в элементах динамически создаваемого связанного списка. Найти в загруженном массиве строк все цепочки символов, являющиеся правильной десятичной записью 16-разрядных двоичных целых чисел без знака.

Билет 18

1. Ссылочные реализации динамических структур данных. Организация списков, операции добавления и исключения элементов. Представление деревьев и графов. Алгоритмы обхода двоичного дерева. Деревья поиска. Примеры использования деревьев в вычислительных алгоритмах.
2. Написать программу, загружающую из текстового файла последовательность целых чисел и заменяющую все элементы исходной последовательности с четным количеством единиц во внутреннем машинном представлении на сумму всех ранее прочитанных элементов. Память для хранения элементов последовательности выделить динамически. Напечатать получившуюся последовательность в порядке возрастания элементов.

Билет 19

1. Реализация множества на базе вектора. Последовательный и двоичный поиск. Битовая реализация множества. Оценка сложности алгоритмов. Хеширование. Методы разрешения коллизий.
2. Написать программу сортировки последовательности целых чисел методом прямого обмена (метод «пузырька»), принимая за ключ сортировки сумму числовых значений старшего и младшего байтов в машинном представлении каждого числа. Последовательность загрузить из файла, память для хранения выделить динамически.

Билет 20

1. Алгоритмы сортировки. Априорная оценка сложности алгоритма. Сравнение эффективности и особенностей реализации различных алгоритмов сортировки.
2. Написать программу загрузки из текстового файла последовательности целых чисел, игнорирующую все элементы исходной последовательности, для которых сумма цифр в их десятичной записи является четным числом. Память выделить динамически на максимальное количество элементов последовательности. Напечатать загруженные элементы в порядке их убывания.

8.2.5. Примеры дополнительных экзаменационных экспресс-задачи (для решения без использования компьютера)

1. Что напечатает программа:

```
#include <stdio.h>
int main(void)
{ short a = 32767;
  short b = 32767;
  printf("%d\n", (short)(a+b));
  return (0);
}
```

Ответ: -2

2. Что напечатает программа:

```
#include <stdio.h>
int main(void)
{ short a = -32768;
  short b = -32768;
  printf("%d\n", (short)(a+b));
  return (0);
}
```

Ответ: 0

3. Что напечатает программа:

```
#include <stdio.h>
int main(void)
{ signed char Low = (signed char)0x80;
  signed char High = (signed char)0x04;
  unsigned short Value;
  Value = ((unsigned short)High << 8) | Low;
  printf("%04x\n", Value);
  return (0);
}
```

Ответ: ff80

4. Что напечатает программа:

```
#include <stdio.h>
int main(void)
{ int a = 635;
  int b = 700;
  printf("%d\n", a + (~b+1));
  return (0);
}
```

Ответ: -65

5. Что напечатает программа:

```
#include <stdio.h>
int main(void)
{ int a[] = { 212, 213, 214, 215, 216, 217, 0 };
  int i;
  for (i = 0; a[i]; ++i)
    printf("%d ", i, a[i] ^ 1);
  printf("\n");
  return (0);
}
```

Ответ: 213 212 215 214 217 216

6. Что напечатает программа:

```
#include <stdio.h>
int main(void)
{ float a = 7;
  printf("%d\n", *(unsigned int*)(&a) >> 23);
  return (0);
}
```

Ответ: 129

7. Что напечатает программа:

```
#include <stdio.h>
int main(void)
{ unsigned int m;
  float a = 7, b;
  m = *(unsigned int*)(&a) & ~(1 << 21);
  b = *(float*)(&m);
  printf("b = %f\n", b);
  return (0);
}
```

Ответ: 6.00000

9. Учебно-методическое и информационное обеспечение дисциплины

1. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. М.: Вильямс, 2007. - 384 с.

2. Бек Л. Введение в системное программирование. М.: Мир, 1988.
3. Валединский В.Д., Пронкин Ю.Н. Вычислительные системы и программирование. В 2-х томах. М.: Изд-во МГУ, 2006. – 344 с.
4. Вирт Н. Алгоритмы и структуры данных. М.: ДМК Пресс, 2010. – 272 с.
5. Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов. М.: Мир, 1981. – 366 с.
6. Керниган Б., Ритчи Д. Язык программирования Си. 2-е изд. М.: Вильямс, 2009. – 304 с.
7. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. 2-е изд. М.: Вильямс, 2002. – 893 с.
8. Липский В. Комбинаторика для программистов. М.: Мир, 1988. – 200 с.
9. Лю Ю-Чжен, Гиббсон Г. Микропроцессоры семейства 8086/8088. М.: Радио и связь, 1987.
10. Морс С.П., Алберт Д.Д. Архитектура микропроцессора 80286. М.: Радио и связь, 1990.
11. Пратт Т., Зелковиц М. Языки программирования: разработка и реализация. 4-е изд. СПб.: Питер, 2002. – 688 с.
12. Седжвик Р. Фундаментальные алгоритмы на С. СПб.: ДиаСофт, 2003. – 1136 с.
13. Таненбаум Э. Архитектура компьютера. 5-е изд. СПб.: Питер, 2007. – 844 с.
14. Таненбаум Э. Современные операционные системы. 2-е изд. СПб.: Питер, 2002. – 1040 с.
15. Уоррен Г. Алгоритмические трюки для программистов. М.: Вильямс, 2004. – 288 с.
16. Deitel Н.М. An Introduction to Operating Systems. 2nd Edition. Addison-Wesley Publishing Company, 1990. – 853 pp.

10. Материально-техническое обеспечение дисциплины

- **Помещения:** компьютерные классы в аудиториях 445 и 446 второго учебного корпуса МГУ.
- **Оборудование:** персональные компьютеры (52 рабочие станции), файловый сервер, сетевое оборудование и необходимое для их работы программное обеспечение.
- **Программное обеспечение рабочих станций:** операционная система Linux, компилятор GNU GCC, файловые менеджеры и тактовые редакторы.

Автор _____

Программа утверждена на заседании кафедры,
протокол № __16__ от __15 мая__2013г.

Заведующий кафедрой
профессор

Г.М. Кобельков