

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)**

Институт информационных технологий и радиоэлектроники

УТВЕРЖДАЮ:
Директор института

Галкин А.А.
« 31 » 08 2021 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ПРОГРАММИРОВАНИЕ

направление подготовки / специальность
09.03.01 Информатика и вычислительная техника

направленность (профиль) подготовки
Системы автоматизированного проектирования микроэлектроники

г. Владимир

2021 Год

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины «Программирование» являются формирование у студентов знаний по основам алгоритмизации вычислительных процессов и общим принципам программирования, представлению основных структур программ и данных.

Задачи:

- изучение основных этапов решения задач на ЭВМ;
- изучение синтаксических и семантических конструкций языков программирования высокого уровня;
- освоение студентами основных приемов программирования типовых задач;
- изучение архитектуры вычислительной машины, принципов обработки машинных команд;
- изучение особенностей реализации обработки различных типов данных;
- освоение студентами соответствия между операторами языков высокого уровня и командами языка ассемблера;
- получение навыков программирования процессора на языке ассемблер.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программирование» относится к обязательной части.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции <i>(код, содержание индикатора)</i>	Результаты обучения по дисциплине	
ОПК-1 Способен применять естественнонаучные и инженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1 Знает основные понятия информатики; принципы программного управления; способы кодирования данных; виды обработки данных ОПК-1.2 Умеет формализовать поставленную задачу, связанную с обработкой данных в рамках заданной предметной области ОПК-1.3 Владеет средствами подготовки, редактирования и оформления текстовой документации, графиков	Знать основные понятия информатики; принципы программного управления; способы кодирования данных; виды обработки данных Уметь формализовать поставленную задачу, связанную с обработкой данных в рамках заданной предметной области Владеть средствами подготовки, редактирования и оформления текстовой документации, графиков	Тестовые вопросы Практико-ориентированное задание
ОПК-2 Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной	ОПК-2.1 Знает современные тенденции развития информационных технологий, вычислительной техники и компьютерных технологий ОПК-2.2 Умеет применять информационные технологии и программные средства для оформления программной документации ОПК-2.3 Владеет текстовыми и графическими редакторами	Знать современные тенденции развития информационных технологий, вычислительной техники и компьютерных технологий Уметь применять информационные технологии и программные средства для оформления программной документации Владеть текстовыми и	Тестовые вопросы Практико-ориентированное задание

деятельности		графическими редакторами	
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1 Знает понятия класс и объект, основные принципы объектно-ориентированного программирования, принципы построения классов, критерии проверки правильности построения классов ОПК-8.2 Умеет использовать системы программирования и предоставляемые пакеты библиотек; выполнять компиляцию, отладку и тестирование составленных программ; разрабатывать основные программные документы ОПК-8.3 Владеет стилями программирования, объектно-ориентированными языками программирования и визуальном программированием, навыками разработки и отладки программ на алгоритмическом языке высокого уровня	Знать понятия класс и объект, основные принципы объектно-ориентированного программирования, принципы построения классов, критерии проверки правильности построения классов Уметь использовать системы программирования и предоставляемые пакеты библиотек; выполнять компиляцию, отладку и тестирование составленных программ; разрабатывать основные программные документы Владеть стилями программирования, объектно-ориентированными языками программирования и визуальном программированием, навыками разработки и отладки программ на алгоритмическом языке высокого уровня	Тестовые вопросы Практико-ориентированное задание

4. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 13 зачетных единиц, 468 часов

Тематический план форма обучения – очная

№ п/п	Наименование тем и/или разделов/тем дисциплины	Семестр	Неделя семестра	Контактная работа обучающихся с педагогическим работником				Самостоятельная работа	Формы текущего контроля успеваемости, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	в форме практической подготовки		
1	Алгоритмизация задач	1	1-3	6	2	4		20	
1.1	Введение. Классификация языков программирования. Этапы создания ПО	1	1	2				4	
1.2	Понятие алгоритма. Способы описания алгоритмов. Принцип пошаговой детализации. Структуры алгоритмов.	1	2	2	2	4		8	

1.3	Система программирования.	1	3	2				8	
2	Синтаксические конструкции	1	4-8	10	6	12		36	
2.1	Понятие алгоритмического языка. С++ Алфавит языка. Переменные и константы. Типы данных	1	4	2	2	4		6	
2.2	Структура программы.	1	5	2				6	
2.3	Линейные программы. Операторы организации разветвляющихся структур	1	6	2	2	4		8	РК1
2.4	Операторы организации циклической обработки	1	7	2				8	
2.5	Неструктурные алгоритмы. Преобразование неструктурных алгоритмов в структурные.	1	8	2	2	4		8	
3	Структурные типы данных	1	9-11	6	2	4		24	
3.1	Массивы. Обработка массивов.	1	9	2				8	
3.2	Символьные массивы. Строки.	1	10	2	2	4		8	РК2
3.3	Структуры.. Перечислимые типы	1	11	2				8	
4	Модульное программирование	1	12-15	8	4	8		28	
4.1	Процедуры и функции	1	12	2	2	4		8	
4.2	Модули	1	13	2				8	
4.3	Файлы. Файловые типы. Обработка компонентов файла	1	14-15	4	2	4		12	
5	Приемы программирования	1	16-18	6	4	8		18	
5.1	Динамические типы данных. Статическая и динамическая память. Указатели. Программирование с использованием динамической памяти	1	16-17	4	2	4		10	
5.2	Задачи сортировки и поиска. Методы сортировки	1	18	2	2	4		8	РК3
Всего за 1 семестр:				36	18	36		126	Экзамен(36)
6	Архитектура и функционирование вычислительной машины	2	1-8	16	14	16		28	
6.1	Представление информации. Программное управление.	2	1	2	2	4		2	
6.2	Уровни рассмотрения машины. Структура вычислительной машины.	2	2	2				5	
6.3	Архитектура учебной ЦВМ. Представление данных. Система команд	2	3	2	2	4		2	
6.4	Программирование для УЦВМ.	2	4-5	4	10	8		5	
6.5	Архитектура IBM PC.	2	6	2				7	
6.6	Регистры процессора. Система прерываний.	2	7	2				2	
6.7	Сегменты.	2	8	2				5	
7	Программирование на языке ассемблера	2	9-18	20	4	20		53	
7.1	Трансляция и компоновка. Инициализация данных. Режимы адресации данных.	2	9-10	4		4		3	
7.2	Команды языка ассемблера	2	11-14	8	2	4		21	

7.3	Подпрограммы.	2	15	2		4		11	
7.4	Приемы программирования.	2	16-17	4		4		7	
7.5	Использование средств языка ассемблера в языках высокого уровня.	2	18	2	2	4		11	
Всего за 2 семестр:				36	18	36		81	Экзамен(45)
Наличие в дисциплине КП/КР									нет
Итого по дисциплине				72	36	72		207	Экзамен(36), Экзамен(45)

Содержание лекционных занятий по дисциплине

1 семестр

Раздел 1. Алгоритмизация задач

Тема 1.1 Введение. Классификация языков программирования. Этапы создания ПО

Классификация языков программирования. Постановка задачи, анализ, проектирование и реализация программного обеспечения.

Тема 1.2 Понятие алгоритма. Способы описания алгоритмов. Принцип пошаговой детализации. Структуры алгоритмов.

Алгоритм, свойства алгоритма. Схема алгоритма, словесное описание алгоритма. Принцип пошаговой детализации при проектировании алгоритма. Структуры линейных, разветвляющихся и циклических вычислительных процессов. Виды циклов: циклы с пред- и постусловием, счетные циклы. Вложенные циклы.

Тема 1.3 Система программирования.

Структура системы программирования: редактор, отладчик, компилятор. Принципы работы в системе программирования. Понятие алгоритмического языка. Синтаксис и семантика языка. Структурный подход.

Раздел 2 Синтаксические конструкции

Тема 2.1 Понятие алгоритмического языка. С++. Алфавит языка. Переменные и константы. Типы данных

Алфавит языка высокого уровня. Основные понятия: идентификаторы, ключевые слова, переменные, константы, метки, типы данных.

Тема 2.2 Структура программы

Структура программы. Стандартные и нестандартные (пользовательские) типы. Простые (базовые) и структурные типы. Перечисляемый тип. Инициализированные переменные. Типизированные константы. Типизированные указатели, ссылочный тип.

Тема 2.3 Линейные программы. Операторы организации разветвляющихся структур

Арифметические и логические выражения. Операции сложения и умножения. Оператор присваивания. Запись линейных программ. Процедуры ввода-вывода. Составной оператор.

Управляющие операторы. Операторы ветвления. Оператор условной передачи управления, оператор выбора

Тема 2.4. Операторы организации циклической обработки

Операторы организации циклической обработки. Операторы основных видов циклов: с предусловием, с постусловием и счетных циклов. Вложенные циклы.

Тема 2.5 Неструктурные алгоритмы. Преобразование неструктурных алгоритмов в структурные

Структурный подход. Неструктурные алгоритмы. Оператор безусловной передачи управления. Операторы неструктурной передачи управления. Методика преобразования неструктурных алгоритмов в структурные

Раздел 3 Структурные типы данных

Тема 3.1 Массивы. Обработка массивов.

Основные понятия: массив, тип элементов массива, размерность массива. Способы объявления и инициализации массива. Операции над массивами: присваивание, доступ к элементам массива (прямая и косвенная адресация), ввод-вывод массивов. Понятие матрицы. Особенности обработки многомерных массивов. Вложенные циклы. Способы обхода элементов матриц.

Тема 3.2 Символьные массивы. Строки.

Понятие символьного массива. Способы объявления и инициализации. Ввод-вывод символьного массива. Операция конкатенации. Переменные строкового типа. Инициализация строк. Операции над переменными строкового типа: доступ к символам строки, присваивание строк, конкатенация строк, операции отношения, ввод-вывод строк. Функции и процедуры для работы со строками.

Тема 3.3 Структуры.. Перечислимые типы

Понятие записи, поля записи. Объявление и инициализация записи. Операции над записями: доступ к полям записи, присваивание записей. Особенности ввода-вывода записей. Оператор присоединения

Раздел 4 Модульное программирование

Тема 4.1 Процедуры и функции.

Процедуры и функции. Назначение, способы оформления подпрограмм. Прототип функции. Область видимости ресурсов, глобальные и локальные переменные. Связь по управлению и связь по данным. Неявная передача данных в подпрограммы – глобальные ресурсы. Явная передача данных в подпрограммы – список параметров. Формальные и фактические параметры. Способы передачи параметров: по значению, по ссылке. Обращение к подпрограммам. Стек. Передача параметров через стек. Схемы алгоритмов. Рекурсия, прямая и косвенная рекурсия. Перегрузка функций. Шаблоны функций

Тема 4.2 Модули.

Основные понятия: модуль, внешние ресурсы модуля, внутренние ресурсы модуля. Структура модуля. Интерфейсная секция, секция реализации, секция инициализации. Подключение модулей. Структура и состав библиотек подпрограмм. Использование библиотек.

Тема 4.3 Файлы. Файловые типы. Обработка компонентов файла

Файлы. Понятие файлового типа. Определение файла. Имя файла, компоненты файла, размер файла, указатель файла. Организация файла. Файловые переменные. Виды файлов: типизированные, нетипизированные, текстовые. Работа с файлами: инициализация файловой переменной, открытие файла, обработка компонентов файла, закрытие файла. Стандартные процедуры и функции обслуживания файлов. Обработка компонентов файла. Понятия последовательного и прямого доступа к компонентам файла.

Раздел 5 Приемы программирования

Тема 5.1 *Динамические типы данных. Статическая и динамическая память. Указатели. Программирование с использованием динамической памяти.*

Динамические типы данных. Статическая и динамическая память. Указатели и операции над ними. Управление динамической памятью: выделение и освобождение памяти.

Тема 5.2 *Задачи сортировки и поиска. Методы сортировки.*

Понятия: эффективность и адекватность метода. Оценка адекватности выбранного метода. Оценка вычислительной сложности алгоритма. Использование библиотечных функций. Задачи поиска и сортировки. Сортировка выбором, включением. Сортировка обменом, шейкер-сортировка, быстрая сортировка Хоара

2 семестр

Раздел 6. Архитектура и функционирование вычислительной машины

Тема 6.1 Представление информации. Программное управление.

Способы представления информации в ЦВМ. Пять принципов программного управления. Пятиблочная машина Фонн-Неймона.

Тема 6.2 Уровни рассмотрения машины. Структура вычислительной машины.

Уровни рассмотрения машины: уровень языков высокого уровня, уровень ассемблера, машинный уровень, уровень микропрограммирования (микроопераций)

Тема 6.3 Архитектура учебной ЦВМ. Представление данных. Система команд.

Архитектура УЦВМ. Представление данных в УЦВМ. Система команд УЦВМ.

Тема 6.4 Программирование для УЦВМ.

Основные шаги при программировании в кодах. Программирование на языке Ассемблер. Организация циклических вычислений и модульных программ.

Тема 6.5 Архитектура IBM PC.

Базовая архитектура IBM PC.

Тема 6.6 Регистры процессора. Система прерываний.

Регистры общего назначения. Регистр флагов. Сегментные регистры. Система прерываний.

Тема 6.7 Сегменты.

Сегменты, сегментные регистры, адресация.

Раздел 7. Программирование на языке ассемблера

Тема 7.1 Трансляция и компоновка. Инициализация данных. Режимы адресации данных.

Трансляция программ на языке ассемблер с использованием TASM.

Тема 7.2 Команды языка ассемблера

Система команд языка ассемблер. Алгоритмы работы команд.

Тема 7.3 Подпрограммы.

Организация связи по данным и по управлению при организации подпрограмм на языке ассемблер.

Тема 7.4 Приемы программирования.

Разбор примеров по организации ввода-вывода и вычисления арифметических выражений.

Тема 7.5 Использование средств языка ассемблера в языках высокого уровня.

Использование языка ассемблер в языках высокого уровня. Организация много модульных программ.

Содержание лабораторных занятий по дисциплине

1 семестр

Раздел 1. Алгоритмизация задач

Лабораторная работа 1 Алгоритмизация линейных вычислительных процессов

Раздел 2 Синтаксические конструкции

Лабораторная работа 2 Алгоритмизация разветвляющихся вычислительных процессов

Лабораторная работа 3 Алгоритмизация циклических вычислительных процессов

Лабораторная работа 4 Программирование вложенных циклов и матричных задач

Раздел 3 Структурные типы данных

Лабораторная работа 5. Алгоритмизация задач обработки символьной информации

Раздел 4 Модульное программирование

Лабораторная работа 6. Процедуры и функции

Лабораторная работа 7. Файлы и структуры

Раздел 5 Приемы программирования

Лабораторная работа 8. Алгоритмизация задач обработки динамических массивов

Лабораторная работа 9. Алгоритмы и программы сортировки

2 семестр

Раздел 6. Архитектура и функционирование вычислительной машины

Тема 6.1 Представление информации. Программное управление.

Практическое занятие 1. Вычисление целевого адреса. Алгоритмизация вычислительного процесса.

Лабораторная работа 1. Программирование на машинном языке.

Тема 6.3 Архитектура учебной ЦВМ. Представление данных. Система команд.

Лабораторная работа 2. Уровень ассемблера

Тема 6.4 Программирование для УЦВМ.

Лабораторная работа 3. Обработка массивов.

Лабораторная работа 4. Организация подпрограмм.

Раздел 7. Программирование на языке ассемблера

Тема 7.1 Трансляция и компоновка. Инициализация данных. Режимы адресации данных.

Лабораторная работа 5. Основы языка Ассемблер МП Intel 8086.

Тема 7.2 Команды языка ассемблера

Лабораторная работа 6. Обработка массивов.

Тема 7.3 Подпрограммы.

Лабораторная работа 7. Организация подпрограмм.

Тема 7.4 Приемы программирования.

Лабораторная работа 8. Вычисление арифметических выражений.

Тема 7.5 Использование средств языка ассемблера в языках высокого уровня.

Лабораторная работа 9. Использование ассемблера в языках высокого уровня

Содержание практических занятий по дисциплине

1 семестр

Раздел 1. Алгоритмизация задач

Практическое занятие 1. Способы описания алгоритмов

Раздел 2 Синтаксические конструкции

Практическое занятие 2. Структура программы на Си. Операторы стандартного ввода/
вывода

Практическое занятие 3. Контрольная работа (РК1)

Практическое занятие 4. Операторы разветвлений и циклов

Раздел 3 Структурные типы данных

Практическое занятие 5. Контрольная работа (РК2).

Практическое занятие 6. Функции работы со строками и символами

Раздел 4 *Модульное программирование*

Практическое занятие 7. Текстовые файлы.

Раздел 5 Приемы программирования

Практическое занятие 8. Алгоритмы сортировки

Практическое занятие 9. Контрольная работа (РК3)

2 семестр

Раздел 6. Архитектура и функционирование вычислительной машины

Тема 6.1 Представление информации. Программное управление.

Практическое занятие 1. Вычисление целевого адреса. Алгоритмизация вычислительного процесса.

Тема 6.3 Архитектура учебной ЦВМ. Представление данных. Система команд.

Практическая работа 2. Программирование на машинном языке.

Тема 6.4 Программирование для УЦВМ.

Практическая работа 3. Контрольная работа (РК1)

Практическая работа 4. Ассемблер УЦВМ.

Практическая работа 5. Обработка массивов. Модификация исполнительного адреса команды.

Практическая работа 6. Обработка массивов. Индексная адресация.

Практическая работа 7. Организация подпрограмм.

Раздел 7. Программирование на языке ассемблера

Тема 7.2 Команды языка ассемблера

Практическая работа 8. Контрольная работа (РК2)

Тема 7.5 Использование средств языка ассемблера в языках высокого уровня.

Практическая работа 9. Контрольная работа (РК9)

5. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

5.1. Текущий контроль успеваемости

1 семестр

Рейтинг-контроль 1

1. В чем состоит назначение компилятора.
2. Приведите классификацию языков программирования по назначению.
3. Опишите отличительные черты структурного и объектно-ориентированного подходов.
4. Перечислите этапы создания программного обеспечения.
5. Дайте понятие алгоритма.
6. Назовите базовые структуры алгоритма.
7. Назовите дополнительные структуры алгоритма.
8. Переменные и константы. Приведите их определения и отличия, назначение.
9. Дайте понятие идентификатора. Для чего он нужен? Как соотносятся переменная и идентификатор?
10. Приведите правила представления (записи) идентификаторов.
11. Что определяет тип переменной?
12. Что такое явное и неявное преобразование типов. Приведите примеры.
13. Выражение. Приведите виды выражений. Опишите приоритеты операций.
14. Как организуются процедуры ввода-вывода? Каковы особенности их использования? Что такое «форматирование вывода на экран»?
15. Опишите оператор присваивания. Что такое «совместимость типов»? Приведите примеры.
16. Опишите структуру программы на языке C++.
17. В каких случаях компилятор выдает сообщение Unknown identifier?
18. В каких случаях компилятор выдает сообщение Type mismatch?
19. Представьте алгоритм решения задачи в виде словесного описания и схемы алгоритма: выведите на экран таблицу умножения, форматированную в виде треугольника.
20. Представьте алгоритм решения задачи в виде словесного описания и схемы алгоритма: выведите на экран первые 20 чисел последовательности Фибоначчи – в 5 рядов друг под другом и по 4 числа в ряду.

Задача:

Составить алгоритм решения задачи и программу на языке C. Требования:

- проверка физического смысла вводимых данных;
- вывод вводимых исходных данных, если они принадлежат области возможных значений, в противном случае вывод сообщения и завершение работы программы.
- вывод результатов работы программы.
- использование стандартных операторов ввода/ вывода.

Вариант 1

Ввести три числа a, b, c , определить могут ли они являться сторонами треугольника. Если треугольник – прямоугольный, то рассчитать площадь, иначе – рассчитать периметр.

Вариант 2

Ввести три числа x, y, R (x, y – координаты точки на плоскости, R – радиус окружности с центром в точке $(0,0)$). Если точка лежит на или внутри окружности, то рассчитать площадь круга, если вне окружности, то длину окружности.

Вариант 3

Ввести три числа v_0 – скорость в нулевой момент времени, a – ускорение, t – время. Определить скорость тела при равноускоренном движении или перемещение, если движение равнозамедленное.

($\mathbf{v} = \mathbf{v}_0 + \mathbf{at}$, $\mathbf{S} = \mathbf{v}_0 t + \mathbf{at}^2/2$, величины \mathbf{v} , \mathbf{v}_0 , \mathbf{a} , \mathbf{S} являются векторными)

Вариант 4

Ввести координаты двух точек x_1, y_1 и x_2, y_2 . Если обе точки принадлежат I или III четверти плоскости, то $z = x_1 + x_2$, иначе $z = (x_1 + x_2)(y_1 + y_2)$.

Вариант 5

Ввести координаты точки x_0, y_0 . Если точка лежит на какой-либо оси, то $z = 0$, иначе $z = x_0^2 y_0^2$.

Вариант 6.

Прямая задана уравнением $y = kx + b$. Ввести коэффициенты прямой k и b , координаты точки x_0, y_0 . Если точка лежит на прямой и функция возрастающая, то $z = x_0 + y_0$, если точка лежит на прямой и функция убывает, то $z = x_0 - y_0$, иначе $z = 0$.

Рейтинг-контроль 2

1. Какие стандартные управляющие структуры используются в структурном программировании для реализации разветвляющихся алгоритмов?
2. Какие управляющие структуры используются в языке C++ для организации разветвляющихся алгоритмов?
3. Приведите синтаксис оператора условной передачи управления и поясните порядок его выполнения (включая его сокращенную форму).
4. Приведите синтаксис оператора выбора и поясните порядок его выполнения.
5. Дайте определение составного оператора. Каким образом он используется в программе?
6. Как разрешается неоднозначность, которая может возникнуть при использовании вложенных операторов `if`?
7. Перечислите виды циклов и приведите их схемы.
8. Приведите функциональную схему счетного цикла и назначение ее отдельных частей.
9. Какие управляющие структуры используются в языке C++ для организации цикла с предусловием?

10. Какие управляющие структуры используются в языке C++ для организации цикла с постусловием?
11. Укажите, сколько операторов можно разместить в теле цикла?
12. Можно ли в теле цикла изменять значение параметра цикла? Ответ пояснить.
13. Можно ли вне тела цикла использовать значение параметра цикла? Ответ пояснить.
14. Дайте понятие массива. Приведите примеры объявления массива в программе всеми известными способами.
15. Опишите принципы работы с одномерными массивами.
16. Опишите принципы работы с многомерными массивами.
17. Дайте определение строкового типа данных. Приведите примеры объявления переменных строкового типа и их инициализацию.
18. Перечислите процедуры и функции для работы со строковым типом данных.
19. Дайте определение символьного массива. В чем состоит отличие символьного массива от переменной строкового типа?
20. Опишите особенности обработки символьного массива.

Задача

Составить алгоритм программы. Написать код на языке C++, реализующие алгоритм с использованием счетного цикла и цикл с предусловием (для нечетных вариантов) и цикл с постусловием (для четных вариантов).

Вариант 1

Сгенерировать матрицу, содержащую целые числа от 0 до 100 размером $N \times M$. Найти сумму и количество чисел, больших 50.

Вариант 2

Сгенерировать матрицу, содержащую целые числа от -100 до 100 размером $N \times M$. Найти сумму и количество чисел отрицательных.

Вариант 3

Сгенерировать матрицу, содержащую целые числа от 0 до 100 размером $N \times M$. Найти произведение и количество чисел, больших 50.

Вариант 4

Сгенерировать матрицу, содержащую целые числа от -10 до 10 размером $N \times M$. Найти произведение и количество чисел отрицательных.

Вариант 5

Сгенерировать матрицу, содержащую целые числа от -100 до 100 размером $N \times M$. Найти номера строк матрицы, в которых встречаются элементы, равные 0 и число нулевых элементов матрицы.

Вариант 6

Сгенерировать матрицу, содержащую целые числа от -100 до 100 размером $N \times M$. Найти номера столбцов матрицы, хотя бы один элемент которых равне 0 и число таких столбцов..

1. Каково назначение подпрограмм? Дайте определение подпрограммы.
2. Назовите виды подпрограмм, их сходства и отличия.
3. Каким образом объявляются процедуры и функции. Что такое прототип функции?
4. Как выглядит вызов (обращение к) процедур и функций в программе?
5. Что такое область видимости переменных? Какова область видимости глобальных и локальных переменных? Приведите поясняющие примеры.
6. В чем состоит назначение локальных переменных?
7. В чем заключается суть явной передачи данных в подпрограмму? Какие возможны варианты явной передачи данных?
8. В чем заключается суть неявной передачи данных в подпрограмму? Какие возможны варианты неявной передачи данных?
9. Что такое список параметров процедуры/функции? Дайте определения формальных и фактических параметров подпрограммы. Приведите примеры.
10. Какие существуют способы передачи параметров в подпрограмму?
11. Опишите механизм передачи параметров по значению.
12. Опишите механизм передачи параметров по ссылке.
13. Каковы достоинства и недостатки вложенных подпрограмм? Какой механизм организации позволяет использовать вложенные подпрограммы?
14. Что такое файловая переменная? Какова ее структура и какую информацию она содержит?
15. Опишите принципы программной обработки файла (основные действия)
16. Перечислите процедуры и функции для работы с текстовыми файлами.
17. Перечислите процедуры и функции для работы с типизированными файлами.
18. Опишите принципы работы с файлами записей.
19. В чем заключаются задачи сортировки и поиска? Какие вы знаете алгоритмы сортировки?
20. Дайте характеристики статической и динамической памяти.
21. Что такое указатель? Для чего он нужен? Приведите примеры.
22. Перечислите основные операции при работе с динамической памятью. Приведите примеры.
Задачи
Составьте алгоритм решения и код на языке C++ (обязательно использование подпрограмм).

Вариант 1

Дана матрица размера $N \times N$. Найти сумму элементов строк матрицы и записать их в отдельном массиве. Подпрограммой оформить суммирование элементов отдельной строки: `<подпрограмма> SumStr (<номер строки матрицы>, <размерность>, ...)`.

Вариант 2

Дана матрица размера $N \times N$, число M . Определить количество элементов со значением M в каждой строке матрицы, результат – в отдельном массиве. Подпрограммой оформить подсчет элементов со значением M в отдельной строке:
`<подпрограмма> NumStr (<строка матрицы>, <размерность>, <значение M>, ...)`.

Вариант 3

Дана матрица размера $N \times N$. Найти число положительных элементов строк матрицы и записать их в отдельном массиве. Подпрограммой оформить подсчет положительных элементов в отдельной строке: `<подпрограмма> NPositiveStr (<номер строки матрицы>, <размерность>, ...)`.

Вариант 4

Дана матрица размера $N \times N$. Заменить все отрицательные элементы матрицы 0 и найти число замен в каждой строке, записать их в отдельном массиве. Подпрограммой оформить подсчет 0-х элементов отдельной строки: <подпрограмма> NNullStr (<номер строки матрицы>, <размерность>,...).

Вариант 5

Дана матрица размера $N \times N$. Найти произведение четных элементов каждой строки, записать их в отдельном массиве. Подпрограммой оформить подсчет произведения четных элементов отдельной строки: <подпрограмма> MulEvenStr (<номер строки матрицы>, <размерность>,...).

Вариант 6

Дана матрица размера $N \times N$. Найти сумму нечетных элементов каждой строки, записать их в отдельном массиве. Подпрограммой оформить подсчет суммы нечетных элементов отдельной строки: <подпрограмма> AddOddStr (<номер строки матрицы>, <размерность>,...).

2 семестр

Рейтинг-контроль 1

1. Опишите алгоритм работы УЦВМ.
2. Перечислите характерные черты языка Ассемблер.
3. Как будут размещены элементы 5-мерного массива в памяти УЦВМ?
4. Что такое структура?
5. Перечислите структурные элементы учебной ЦВМ.
6. Назовите режимы адресации в УЦВМ.
7. В каком коде представлены числа в УЦВМ?
8. Значение какого регистра используется при индексной адресации?
9. Перечислите команды, изменяющие естественный порядок выполнения программы.
10. Сколько принципов программного управления Вы знаете?
11. Назовите минимальную единицу информации.
12. Назовите минимальную адресуемую единицу информации.
13. Сколько форматов команд существует в УЦВМ?
14. Сколько программно доступных регистров существует в УЦВМ?

Рейтинг-контроль 2

Вариант 1

1. Назначение регистров в учебной машине
2. Как можно поменять местами значения в регистрах L и X
3. Какие команды меняют признак результата
4. Для чего нужны псевдокоманды start и end
5. Сколько байт в памяти занимает команда clear
6. Написать на Ассемблере учебной машины программу, выполняющую алгоритм, представленный на рисунке.

Вариант 2

1. Форматы данных в учебной машине
2. Как произвести действие $A=B-C$ без использования команды sub

3. Какие команды меняют естественный порядок выполнения программы
4. Для чего нужны псевдокоманды `word` и `resb`
5. Сколько байт в памяти занимает команда `pop`
6. Написать на Ассемблере учебной машины программу, выполняющую алгоритм, представленный на рисунке.

Рейтинг-контроль 3

Вариант 1

1. Охарактеризуйте модели памяти `tiny` и `small`. Чем они отличаются?
2. Сколько байт в памяти займет переменная:
 - 2.1 `S11 db 'Hii'`
 - 2.2 `b dt 4`
 - 2.3 `Sdr dd 2 dup (5, 2, 8)`
3. Сколько символов выведется на экран после выполнения фрагмента программы, представленного на рисунке 1.
 - 4.1 Команда **`pop`**.
 - 4.2 Команда **`loopz`**.
5. Написать команду в результате выполнения которой значение в регистре `CX` увеличится в 4 раза (действие должно быть реализовано за одну команду)
6. Что будет в регистре `CX` после выполнения команды:
`and cx,cx`
7. Написать на ассемблере программу для реализации алгоритма эквивалентного представленному на языке `C++` на рисунке 2.

Вариант 2

1. Охарактеризуйте модели памяти `large` и `huge`. Чем они отличаются?
2. Сколько байт в памяти займет переменная:
 - 2.1 `S1 db 'Hello'`
 - 2.2 `A dq ?`
 - 2.3 `Sdrt dw 4 dup (1, 2)`
3. Сколько символов выведется на экран после выполнения фрагмента программы, представленного на рисунке 1.
 - 4.1 Команда **`push`**.
 - 4.2 Команда **`loopne`**.
5. Написать команду в результате выполнения которой значение в регистре `BX` уменьшится в 16 раз (действие должно быть реализовано за одну команду)
6. Что будет в регистре `CX` после выполнения команды:
`xor cx,cx`
7. Написать на ассемблере программу для реализации алгоритма эквивалентного представленному на языке `C++` на рисунке 2.

5.2. Промежуточная аттестация

Вопросы на экзамен (семестр 1)

1. Понятие алгоритма. Требования к алгоритму. Свойства алгоритма.
2. Структуры алгоритмов. Способы записи алгоритмов.
3. Этапы решения задач на ЭВМ.
4. Принципы структурного программирования.
5. Типы данных и константы языка `C++`.

6. Типизированные указатели.
7. Ссылочный тип.
8. Перечислимый тип.
9. Структура программы на языке C++.
10. Процедуры ввода-вывода. Поточковый ввод-вывод.
11. Оператор присваивания. Основные арифметические операции C++.
12. Выражения. Основные операции языка C++ .
13. Условный оператор. Оператор выбора.
14. Операторы безусловного перехода.
15. Циклические структуры языка C++. Виды циклов.
16. Циклические программы. Вложенные циклы.
17. Глобальные и локальные переменные.
18. Функции. Механизм передачи параметров.
19. Вложенные функции. Рекурсия.
20. Область видимости и время жизни переменной.
21. Фактические и формальные параметры.
22. Перегрузка функций.
23. Шаблоны функций.
24. Понятие модуля. Преимущества модульного программирования.

Структура модуля.

25. Пример модуля. Способ использования модуля.
26. Массивы в языке C++. Связь указателей и массивов в C++.
27. Символьный тип данных. Строковые массивы. Способы обработки.
28. Основные операции над строками. Функции для работы со строками.
29. Структуры. Работа со структурами. Примеры.
30. Массивы структур. Особенности обработки. Примеры.
31. Файлы. Виды файлов. Файловая переменная. Общая схема работы с файлами.
32. Файлы. Виды файлов. Общие для всех файлов функции обработки.
33. Текстовые файлы. Функции обработки для текстовых файлов.
34. Бинарные файлы. Функции для работы с бинарными файлами.
35. Статическая и динамическая память.
36. Работа с динамической памятью. Примеры.
37. Алгоритмы и методы сортировки: оценка эффективности алгоритма.
38. Сортировка выбором.
39. Сортировка включением.
40. Сортировка обменом.
41. Шейкер-сортировка.
42. Быстрая сортировка Хоара.
43. Алгоритмы и методы поиска в отсортированном массиве данных.

Вопросы на экзамен (семестр 2)

1. Принципы программного управления
2. Пятиблочная машина фон Неймана. Понятия низкоуровневого программирования.

3. Понятие архитектура ВМ. Составные части архитектуры.
4. Схема процесса разработки программы на ассемблере
5. Базовая архитектура IBM PC X86. Представление данных с точки зрения размерности.
6. Базовая архитектура IBM PC X86. Представление данных с точки зрения интерпретации
7. Базовая архитектура IBM PC X86. Регистр флагов.
8. Базовая архитектура IBM PC X86. Регистры общего назначения. AX, BX, CX, DX.
9. Базовая архитектура IBM PC X86. Регистры общего назначения. SI, DI, BP, SP, IP.
10. Базовая архитектура IBM PC X86. Сегментные регистры. CS, DS, ES, SS
11. Базовая архитектура IBM PC X86. Сегменты. Адресация.
12. Базовая архитектура IBM PC X86. Структура команд
13. Базовая архитектура IBM PC X86. Система команд (виды команд)
14. Базовая архитектура IBM PC X86. Система прерываний. Организация ввода-вывода.
15. Определение «ассемблер». Характерные черты Ассемблера.
16. Схем трансляции.
17. Ассемблер Intel 8086. Синтаксис: программы, строки, комментария.
18. Ассемблер Intel 8086. Синтаксис: команды, метки.
19. Ассемблер Intel 8086. Синтаксис: директивы, макрокоманды.
20. Ассемблер Intel 8086. Сегментные директивы. Модели памяти.
21. Ассемблер Intel 8086. Режимы адресации данных
22. Ассемблер Intel 8086. Инициализации данных (директивы, примеры).
23. Ассемблер Intel 8086. Именованные области памяти
24. Ассемблер Intel 8086. Индексирование элементов массивов, включая многомерных
25. Ассемблер Intel 8086. Способы обращения к элементам массива.
26. Ассемблер Intel 8086. Команды пересылки данных. Mov, xchg, lea, lds, les,
27. Ассемблер Intel 8086. Команды пересылки данных. Push, pop, pushf, popf, lahf, sahf
28. Ассемблер Intel 8086. Команды пересылки данных. Xlat, in, out.
29. Ассемблер Intel 8086. Арифметические операции. Add, adc, sub, sbb, mul, imul, div, idiv
30. Ассемблер Intel 8086. Арифметические операции. Cmp, cbw, cwd, dec, inc, neg
31. Ассемблер Intel 8086. Арифметические операции. Aaa, aad, aam, aas, daa, das
32. Ассемблер Intel 8086. Команды передачи управления. Jmp, j?*, jcjx
33. Ассемблер Intel 8086. Команды передачи управления. Loop, loope, loopz, loopne, loopnz
34. Ассемблер Intel 8086. Команды передачи управления: call, ret. Команды работы с битами: test
34. Ассемблер Intel 8086. Логические операции. AND, NOT, OR, XOR
36. Ассемблер Intel 8086. Команды работы с битами. Sal, shl, sar, shr
37. Ассемблер Intel 8086. Команды работы с битами. Rcl, rcr, rol, ror
38. Ассемблер Intel 8086. Команды обработки строк. CMPS, CMPSB, CMPSW
39. Ассемблер Intel 8086. Команды обработки строк. LODS, LODSB, LODSW
40. Ассемблер Intel 8086. Команды обработки строк. MOVS, MOVSB, MOVSW
41. Ассемблер Intel 8086. Команды обработки строк. SCAS, SCASB, SCASW
42. Ассемблер Intel 8086. Команды обработки строк. STOS, STOSB, STOSW
43. Ассемблер Intel 8086. Команды обработки строк: префиксы команд. REP, REPE, REPZ, REPNE, REPNZ
44. Ассемблер Intel 8086. Команды изменения состояния процессора. CLC, CLD, CLI, CMC, STC, STI
45. Ассемблер Intel 8086. Команды изменения состояния процессора. ESC, HLT, LOCK, NOP, WAIT
46. Ассемблер Intel 8086. Команды для работы с прерываниями. INT, INTO, IRET
47. Организация ввода символа и строки на экран средствами DOS в ассемблер Intel 8086
48. Организация вывода символа и строки на экран средствами DOS в ассемблер Intel 8086
49. Вычисление выражений в ассемблере.
50. Теоретические основы многоразрядного сложения и вычитания.

51. Теоретические основы многоуровневого умножения.
52. Теоретические основы преобразования числа в строку.
53. Теоретические основы преобразования строки в число.
54. Организация подпрограмм. Определение подпрограммы. Назначение подпрограммы. Проблемы, возникающие при организации подпрограмм. Связь по управлению.
55. Связь программы и подпрограммы по данным. Рекомендуемый порядок оформления подпрограммы
56. Ассемблер Intel 8086. Определение идентификаторов
57. Ассемблер Intel 8086. Директивы для организации многомодульных программ.
58. Ассемблер Intel 8086. Сегментные директивы
59. Формы комбинирования программ на языках высокого уровня с ассемблером.
60. Основные соглашения по передаче параметров в процедуру.
61. Тонкости вызова методов между C++ и Ассемблером

5.3. Самостоятельная работа обучающегося.

Целью самостоятельной работы являются формирование личности студента, развитие его способности к самообучению и повышению своего профессионального уровня.

Самостоятельная работа заключается в изучении содержания тем курса по конспектам, учебникам и дополнительной литературе, подготовке к лабораторным и практическим занятиям, выполнении заданий для самостоятельной работы, оформлении отчетов по лабораторным работам, подготовке к промежуточной аттестации.

Задания для самостоятельной работы студентов (семестр 1)

Раздел 1.

Составить ответы на следующие вопросы

1. Какие существуют способы записи алгоритма?
2. Метод нисходящего проектирования, в чем его особенности?

Задача 1

Составить словесный алгоритм

- | | |
|-----------|--|
| Вариант 1 | Нахождение корней линейного уравнения $kx+b=0$. |
| Вариант 2 | Нахождение корней квадратного уравнения $ax^2 + c=0$. |
| Вариант 3 | Нахождение корней кубического уравнения $ax^3 + c=0$. |
| Вариант 4 | Определить, являются ли числа a, b, c сторонами остроугольного треугольника. |
| Вариант 5 | Определить, являются ли числа a, b, c сторонами прямоугольного треугольника. |
| Вариант 6 | Определить, являются ли числа a, b, c сторонами тупоугольного треугольника. |

Раздел 2.

Составить ответы на следующие вопросы

- 1) Какие существуют способы записи алгоритма?
- 2) Метод нисходящего проектирования, в чем его особенности?
- 3) Линейные операторы языка C++.
- 4) Простые типы данных языка C++.
- 5) Структура программы на языке C++.
- 6) Стандартные библиотеки и их подключение.
- 7) Что такое идентификатор, переменная, константа?

- 8) Что такое совместимость типов?
- 9) Явное и неявное преобразование типов.
- 10) Какие управляющие структуры используются в языке C++ для организации разветвляющихся алгоритмов?
- 11) Поясните порядок выполнения оператора *if...else* и его сокращенной формы?
- 12) Поясните порядок выполнения оператора *switch*. Каково здесь назначение оператора *break*?
- 13) Сколько операторов можно написать после ключевых слов *if* и *else*?
- 14) Что такое составной оператор?
- 15) Как разрешается неоднозначность, которая может возникнуть при использовании вложенных операторов *if*?
- 16) Функциональная схема цикла и назначение ее отдельных частей.
- 17) Какие виды циклов существуют, в чем их принципиальные отличия?
- 18) Назначение цикла *for* и его отдельных компонентов.
- 19) Укажите, сколько операторов можно разместить в теле цикла?
- 20) Можно ли в теле цикла изменять значение параметра цикла?
- 21) Можно ли вне тела цикла использовать значение параметра цикла?
- 22) Как работают операторы *while* и *do-while*?
- 23) Какими способами можно инициализировать массив?
- 24) Какими способами можно обращаться к элементам массива?
- 25) Какова общая схема работы с массивом?

Задача 1.

Составить программу, формирующую таблицу умножения, и выводящую результат в виде верхнетреугольной матрицы. Использовать вложенные счетные циклы.

Задача 2.

Дан массив из N действительных чисел. Составить структурный алгоритм для нахождения :

- 1) Первого отрицательного элемента массива. Вывести элемент и его номер в массиве.
- 2) Последнего отрицательного элемента массива. Вывести элемент и его номер в массиве.
- 3) Элемента массива, который является квадратом целого числа. Вывести элемент и его номер в массиве.
- 4) Элемента массива, который является четным целым числом. Вывести элемент и его номер в массиве.
- 5) Сумму элементов с четными номерами. Вывести сумму.
- 6) Числа положительных, отрицательных и нулевых членов массива.

Раздел 3.

Составить ответы на следующие вопросы

- 1) Какова общая схема работы с многомерным массивом?
- 2) Каковы правила организации вложенных циклов?
- 3) Какие существуют способы объявления массива?
- 4) Назовите алгоритм работы и правила использования неструктурного оператора *break*.
- 5) Какие существуют способы организации символьных данных?
- 6) Объявление и инициализация символьных массивов.
- 7) В чем отличие символьной и строковой константы?

- 8) Как хранится строковое значение в памяти ЭВМ?
- 9) Общие принципы работы с символьными данными.
- 10) Библиотеки и функции для работы со строками.
- 11) Какие возможны операции над строками?

Задача 1.

Разработать алгоритм для кодирования двоичной последовательности методом Лемпела-Зива. Исходную последовательность обрабатывать в виде строки. При реализации программы использовать библиотечные функции работы с символами и строками. Результатом работы должно быть: 1) длина кодового слова в словаре фраз, 2) кодовые слова, представленные массивом строк, каждая из которых является кодом фразы словаря.

Задача 2.

(Обратная задаче 1). Разработать алгоритм для декодирования данных, закодированных по методу Лемпела-Зива. Исходные данные 1) длина кодового слова в словаре фраз, 2) количество кодовых слов, 3) последовательность кодовых слов, вводимых с клавиатуры (при тестировании использовать результаты работы задачи 1). В программе предусмотреть вариант некорректного ввода кодового слова с выдачей информационного сообщения. Результат представить в виде одной строки.

Раздел 4.

Составить ответы на следующие вопросы

- 1) Что такое подпрограмма? Каково ее назначение?
- 2) Назовите основные принципы модульности и как они реализованы в C++.
- 3) Что такое область видимости и время жизни объекта программы?
- 4) Глобальные и локальные переменные.
- 5) Формальные и фактические параметры.
- 6) Назовите основные принципы структурного программирования.
- 7) Как происходит передача параметров в подпрограмму, какие аппаратные и программные средства для этого используются?
- 8) В чем принципиальное отличие передачи параметра по значению и по ссылке, в каких случаях используется одно и другое?
- 9) Что такое синоним переменной, чем он отличается от указателя?
- 10) Для чего нужно предварительное объявление функции?
- 11) Что такое файл? Общая схема работы с файлом.
- 12) Что содержит файловая переменная?
- 13) Как работают функции *fopen* и *fclose*?
- 14) Что такое структура? Как обращаться к полям структуры?
- 15) Как выглядит объявление структуры? Как в памяти хранится переменная структурного типа?
- 16) В чем отличие обработки текстового и бинарного файлов?

Задача 1.

Разработать программу кодирования и декодирования данных по методу Лемпела-Зива. Кодруемая/декодированная последовательность и кодовые слова содержатся в :
1) текстовых файлах;

2) двоичных файлах.

Задача 2.

Реализовать алгоритм кодирования/ декодирования по методу Лемпела-Зива с использованием функций. В виде функций реализовать следующие алгоритмы: 1) поиск фразы-прототипа для новой фразы, результат – новая фраза, код фразы, 2) формирование фразы при декодировании, результат – новая фраза.

Раздел 5.

Составить ответы на следующие вопросы

- 1) Какие виды памяти Вы знаете? Каково их назначение?
- 2) Что такое динамическая память? В чем заключается ее особенность? Правила работы с динамической памятью.
- 3) Как работают операторы *new* и *delete*?
- 4) Что такое шаблон функции? Для чего он используется?
- 5) На каком механизме базируется реализация шаблонов функций?
- 6) Что такое параметризация шаблона? Чем он параметризируется?
- 7) В чем заключается параметризация шаблона?
- 8) Что такое «метод дихотомии»? Для чего он используется?
- 9) Как оценить эффективность того или иного метода сортировки?
- 10) Для чего используется сортировка? В каких случаях затраты на сортировку оправданы?
- 11) Как осуществляется добавление новых данных в отсортированный массив?

Задания для самостоятельной работы студентов (семестр 2)

Раздел 6

1. Для каждой из предложенных задач записать алгоритм решения задачи на C++; схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; распределение памяти и текст программы на машинном языке УЦВМ (Запись алгоритма на C++ и блок-схема должна быть согласованы между собой и составлены таким образом, чтобы машинная программа УЦВМ была их реализацией на низком уровне)
 - а) Даны три переменные. Определить наибольшее значение.
 - б) Даны три переменные. Результату присвоить значение 1, если переменные упорядочены по возрастанию; иначе - 0.
 - в) Определить n-й член ряда Фибоначчи: $F[1]=1$, $F[2]=1$, $F_n=F[n-1]+F[n-2]$.
 - г) Вычислить $y = 1 + x + 2*x^2 + 3*x^3 + 4*x^4 + 5*x^5$.
 - д) Известно, что две переменные имеют одинаковые значения, а третья - отличное от них. Найти ее значение.
 - е) Составить программу перемножения двух чисел без применения команды умножения.
 - ж) Составить программу целочисленного деления двух чисел без применения команды деления. Если деление невозможно, то переменной, называемой индикатором ошибки, присвоить значение 1, иначе - 0.
 - з) Составить программу для нахождения наибольшего общего делителя двух чисел.
 - и) Вычислить факториал числа n, где $n \geq 1$.
 - к) Вычислить n-ю степень числа, где $n \geq 1$.

- л) Определить максимальное число, факториал которого может быть представлен в машине .
м) Вычислить $y = \min(x^2, \max(y, 10))$.
н) Вычислить число размещений из m по n по формуле $A(n, m) = m(m-1)(m-2)\dots(m-(n-1))$, где $m \geq n$.

34

- о) Вычислить $y = \max(x-1, y^2, z \div 3)$.
п) Даны три переменные с неравными значениями, рассматриваемые как координаты точек на числовой оси. Определить длину минимального отрезка, включающего все три точки.

2. Составить на языке Ассемблер учебной ЦВМ программу решения каждой задачи из задания 1, провести ее ассемблирование, изучить полученную при этом программу на машинном языке.

3. Для каждой из предложенных задач записать алгоритм решения задачи на C++ (с использованием массивов); схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; текст программы на языке Ассемблер УЦВМ (Запись алгоритма на C++ и блок-схема должны быть согласованы между собой и составлены таким образом, чтобы программа для УЦВМ была их реализацией на низком уровне)

- а) Найти значение минимального элемента массива.
б) Найти индекс максимального элемента массива.
в) Найти скалярное произведение двух векторов.
г). Найти сумму элементов главной диагонали квадратной матрицы.
д) Найти сумму положительных элементов массива.
е) Найти индекс элемента с заданным значением. Предполагается, что в массиве есть такой элемент и он единственный.
ж) Определить количество положительных и отрицательных элементов массива.
з) Определить количество элементов массива, которым предшествуют элементы с меньшими значениями.
и) Каждому элементу массива, начиная со второго, присвоить значение максимального элемента из числа ему предшествующих и его самого.
к) Дана прямоугольная матрица. Найти сумму элементов строки с заданным номером.
л) Определить, какие два последовательных элемента массива наименее отличаются друг от друга. Найти индекс первого элемента пары.
м) Построить массив, элементы которого суть суммы последовательных пар элементов исходного массива.
н) Определить количество элементов массива, значения которых превышают заданное.
о) Массив, элементы которого принадлежат множеству $\{0, 1\}$, рассматривается как представление целого числа. Определить значение числа, заданного таким способом.
п) Дан массив, элементы которого принадлежат множеству $\{0, 1\}$. Определить длину первой последовательности рядом стоящих единиц.

4. Для каждой из предложенных в задании 1 задач записать алгоритм решения задачи на C++ с использованием подпрограмм; схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; текст программы на языке Ассемблер УЦВМ (Запись алгоритма на C++ и блок-схема должны быть согласованы между собой и составлены таким образом, чтобы программа для УЦВМ была их реализацией на низком уровне)

Раздел 7.

1. Для каждой из предложенных задач записать алгоритм решения задачи на C++; схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; текст программы на языке ассемблер (реальной ЭВМ) (Запись алгоритма на C++ и блок-схема должны быть согласованы между собой и составлены таким образом, чтобы программа на языке ассемблер)

- а) Даны три переменные. Определить наибольшее значение.
- б) Даны три переменные. Результату присвоить значение 1, если переменные упорядочены по возрастанию; иначе - 0.
- в) Определить n-й член ряда Фибоначчи: $F[1]=1$, $F[2]=1$, $F_n=F[n-1]+F[n-2]$.
- г) Вычислить $y = 1 + x + 2*x^2 + 3*x^3 + 4*x^4 + 5*x^5$.
- д) Известно, что две переменные имеют одинаковые значения, а третья - отличное от них. Найти ее значение.
- е) Составить программу перемножения двух чисел без применения команды умножения.
- ж) Составить программу целочисленного деления двух чисел без применения команды деления. Если деление невозможно, то переменной, называемой индикатором ошибки, присвоить значение 1, иначе - 0.
- з) Составить программу для нахождения наибольшего общего делителя двух чисел.
- и) Вычислить факториал числа n, где $n \geq 1$.
- к) Вычислить n-ю степень числа, где $n \geq 1$.
- л) Определить максимальное число, факториал которого может быть представлен в машине.
- м) Вычислить $y = \min(x^2, \max(y, 10))$.
- н) Вычислить число размещений из m по n по формуле $A(n, m) = m(m-1)(m-2)\dots(m-(n-1))$, где $m \geq n$.
- о) Вычислить $y = \max(x-1, y^2, z \text{ div } 3)$.
- п) Даны три переменные с неравными значениями, рассматриваемые как координаты точек на числовой оси. Определить длину минимального отрезка, включающего все три точки.

2. Для каждой из предложенных задач записать алгоритм решения задачи на C++ (с использованием массивов); схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; текст программы на языке ассемблер (реальной ЭВМ) (Запись алгоритма на C++ и блок-схема должна быть согласованы между собой и составлены таким образом, чтобы программа для УЦВМ была их реализацией на низком уровне)

- а) Найти значение минимального элемента массива.
- б) Найти индекс максимального элемента массива.
- в) Найти скалярное произведение двух векторов.
- г). Найти сумму элементов главной диагонали квадратной матрицы.
- д) Найти сумму положительных элементов массива.
- е) Найти индекс элемента с заданным значением. Предполагается, что в массиве есть такой элемент и он единственный.
- ж) Определить количество положительных и отрицательных элементов массива.
- з) Определить количество элементов массива, которым предшествуют элементы с меньшими значениями.
- и) Каждому элементу массива, начиная со второго, присвоить значение максимального элемента из числа ему предшествующих и его самого.
- к) Дана прямоугольная матрица. Найти сумму элементов строки с заданным номером.
- л) Определить, какие два последовательных элемента массива наименее отличаются друг от друга. Найти индекс первого элемента пары.
- м) Построить массив, элементы которого суть суммы последовательных пар элементов исходного массива.
- н) Определить количество элементов массива, значения которых превышают заданное.
- о) Массив, элементы которого принадлежат множеству $\{0,1\}$, рассматривается как представление целого числа. Определить значение числа, заданного таким способом.
- п) Дан массив, элементы которого принадлежат множеству $\{0,1\}$. Определить длину первой последовательности рядом стоящих единиц.

3. Для каждой из предложенных в задании 1 задач записать алгоритм решения задачи на C++ (с использованием подпрограмм); схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; текст программы на языке ассемблер (реальной ЭВМ) (Запись алгоритма на C++ и блок-схема должны быть согласованы между собой и составлены таким образом, чтобы программа для УЦВМ была их реализацией на низком уровне)

4. Составить текст программы на языке ассемблер (реальной ЭВМ) для следующих задач:

а) Объединить два файла с записью их содержимого в третий (новый) файл.

б) Подсчитать побайтовую контрольную сумму файла (4 байта, переполнение результата игнорируется). Записать полученное значение в новый файл.

в) Сгенерировать бинарный файл, компонентами которого являются значения функции $n!$; n меняется от 0 до 10 с шагом 1. Для представления числа использовать 2 байта (слово).

г) Даны два файла одинакового размера. Получить третий файл, содержащий разности между компонентами первого и второго файлов. Размер компонентов - 1 байт. Переполнение результата игнорируется.

д) Дан текстовый файл. Получить новый файл, удалив все символы с кодом 10 из исходного файла.

е) Скопировать файл, используя буфер размером 1 килобайт. На размер файла ограничений не накладывается.

ж) Дан текстовый файл. Подсчитать количество вхождений подстроки «abc» с записью результата (2 байта) в новый файл.

з) Дан файл, размер которого не превышает одного килобайта. Разработать программу, производящую инвертирование компонент файла. Размер компонент - 1 байт. Результат записывается поверх существующей информации в том же файле.

и) Разбить файл на две равные по размеру части, создав два новых файла. Для получения размера файла использовать подпрограмму `file_seek`.

к) Разбить файл на две части, записав байты с нечетными номерами в один выходной файл, а с четными - в другой.

л) Дан текстовый файл. Получить новый файл, заменив символы «1» в исходном файле на «2» и наоборот.

м) Получить файл, отличающийся от исходного обратным порядком следования компонент. Размер компонент - 1 байт.

н) Дан текстовый файл. Получить файл на основе исходного, приведя все символы, соответствующие латинским буквам, к верхнему регистру.

о) Дан текстовый файл, представляющий собой программу на языке ассемблер. Получить файл на основе исходного, удалив все комментарии.

п) Создать в каталоге 50 пустых файлов с именами «file_???.txt», где ?? - порядковый номер файла. Учтите, что по умолчанию приложения MS-DOS не могут одновременно работать более чем с 16 файлами.

Фонд оценочных материалов (ФОМ) для проведения аттестации уровня сформированности компетенций обучающихся по дисциплине оформляется отдельным документом.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Книгообеспеченность

Наименование литературы: автор, название, вид издания, издательство	Год издания	КНИГООБЕСПЕЧЕННОСТЬ
		Наличие в электронном каталоге ЭБС
Основная литература		
1. Дейл, Н. Программирование на C++ / Дейл Н., Уимз Ч., Хедингтон М. - Москва : ДМК Пресс, 2007. - 672 с.	2007	https://www.studentlibrary.ru/book/ISBN5937000080.html

(Серия "Учебник") - ISBN 5-93700-008-0		
2. Секаев, В. Г. Основы программирования на Ассемблере : учеб. пособие / Секаев В. Г. - Новосибирск : Изд-во НГТУ, 2010. - 100 с. - ISBN 978-5-7782-1473-6	2010	https://www.studentlibrary.ru/book/ISBN9785778214736.html
3. Коломиец И.А. С++ : методические указания к лабораторным работам / И. А. Коломиец , С. С. Гладько ; ВлГУ, Кафедра вычислительной техники .— Владимир : Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых (ВлГУ), 2012 .— 38 с.	2012	http://e.lib.vlsu.ru/bitstream/123456789/2779/1/00284.pdf
4. Сущинина А.А. Методические указания к лабораторным работам по дисциплине «Программирование» [Электронный ресурс]/ А. А. Сущинина; ВлГУ, Кафедра вычислительной техники и систем управления.— Владимир, ВлГУ, 2018 .— 22 с.	2018	http://e.lib.vlsu.ru/bitstream/123456789/6905/1/00740.doc
Дополнительная литература		
1. Головинов И.А. Программирование на языке Ассемблера i8086 : практикум / И. А. Головинов .— Владимир : ВлГУ, 2003 .— 70 с.	2003	http://e.lib.vlsu.ru:80/handle/123456789/699
2. Медведева О.Н. Программирование [Электронный ресурс] : курс лекций / О. Н. Медведева - Владимир : ВлГУ, 2011 .— 145 с.	2011	http://e.lib.vlsu.ru/bitstream/123456789/3090/1/00650.pdf

6.2. Периодические издания

Журналы (<https://elibrary.ru/>):

1. Вестник компьютерных и информационных технологий
2. Вычислительные технологии
3. Известия вузов: электроника
4. Радиотехнические и телекоммуникационные системы

6.3. Интернет-ресурсы

<http://www.studentlibrary.ru>

<http://library.vlsu.ru/>

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации данной дисциплины имеются специальные помещения для проведения занятий *лекционного типа, занятий практического/лабораторного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы.* Практические/лабораторные работы проводятся в аудиториях 401-2, 412-2 и 416-2.

Рабочую программу составили:

доцент кафедры ВТ и СУ Калыгина Л.А.
ст. преподаватель кафедры ВТ и СУ Сущинина А.А.



Рецензент

(представитель работодателя)

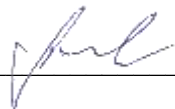


Генеральный директор ООО "Диаграмма" Протягов И.В.

Программа рассмотрена и одобрена на заседании кафедры ВТ и СУ

Протокол № 1 от 31 августа 2021 года

Заведующий кафедрой Ланцов В.Н.

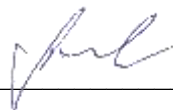


Рабочая программа рассмотрена и одобрена

на заседании учебно-методической комиссии направления 09.03.01 информатика и
вычислительная техника

Протокол № 1 от 31 августа 2021 года

Председатель комиссии Ланцов В.Н. зав. каф. ВТиСУ



**ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ
РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ**

Рабочая программа одобрена на 20 22 / 20 23 учебный года

Протокол заседания кафедры № 1 от 29.08.22 года

Заведующий кафедрой  Кузнецов К.В.

Рабочая программа одобрена на 20 ____ / 20 ____ учебный года

Протокол заседания кафедры № ____ от ____ года

Заведующий кафедрой _____

Рабочая программа одобрена на 20 ____ / 20 ____ учебный года

Протокол заседания кафедры № ____ от ____ года

Заведующий кафедрой _____