

Введение

Информатика - это техническая наука, систематизирующая приемы создания, хранения, обработки и передачи данных средствами вычислительной техники.

Информатика изучает общие закономерности, свойственные информационным процессам. При разработке новых носителей информации, каналов связи, приемов кодирования, визуального отображения информации важен процесс передачи и сохранения информации. Для разработчика системы управления базами данных (СУБД) важны общие принципы организации, эффективность поиска данных и объемы хранимой информации.

Многообразные информационные технологии, функционирующие в разных видах человеческой деятельности (управлении производственным процессом, проектировании, финансовых операциях, образовании и т.п.) существенно различаются между собой.

В настоящее время ни одна отрасль науки и техники не развивается такими темпами, как вычислительная техника. Широкое внедрение средств вычислительной техники (ВТ), а также автоматизация, компьютеризация и информатизация во всех сферах человеческой деятельности; насыщенность современной науки и производства средствами ВТ требует грамотной эксплуатации подобной техники. Сегодня молодой специалист должен обладать хорошими знаниями самих средств ВТ, принципов их работы, математического и программного обеспечения современных ЭВМ.

Информатика решает следующие задачи:

- разработка аппаратных средств и архитектуры вычислительных систем;
- программирование (приемы, методы и средства разработки компьютерных программ);
- разработка интерфейсов вычислительных систем (приемов и методов взаимодействия с аппаратным и программным обеспечением);
- преобразование данных;
- защита информации;
- автоматизация (функционирование средств ВТ без участия человека);
- стандартизация (обеспечение совместимости между аппаратными и программными средствами, а также между форматами представления данных, относящихся к различным типам вычислительных систем).

Дисциплина "Информатика" является общеобразовательной дисциплиной, входящей в цикл общих естественнонаучных дисциплин и изучается студентами всех специальностей заочной формы обучения.

Знания, навыки и умения, полученные студентами в процессе изучения дисциплины, обеспечивают более успешное изучение последующих учебных дисциплин, способствует качественному выполнению курсовых работ и дипломного проекта.

Предлагаемое пособие состоит из двух частей. Первая часть включает содержание дисциплины, методические указания по ее изучению и вопросы для самопроверки. Вторая часть включает варианты контрольной работы и методические указания по ее выполнению.

Часть I. Методические указания по изучению дисциплины Общие положения по организации процесса обучения

Для студентов первого курса заочной формы цикл обучения по дисциплине "Информатика" предусматривает:

Общий объём учебных часов на дисциплину	<i>180 ч</i>
Лекции	<i>10 ч</i>
Лабораторные занятия	<i>8 ч</i>
Практические занятия	<i>1 ч</i>
Самостоятельная работа	<i>160 ч</i>
Курсовой проект (курсовая работа)	<i>нет</i>
Экзамен	<i>1 семестр</i>
Зачёт	<i>нет</i>
Контрольная работа	<i>2</i>

1. Цели освоения дисциплины

Целью освоения дисциплины является формирование необходимых знаний в области аппаратного и программного обеспечения персональных компьютеров, алгоритмизации и программирования, а также развитие навыков по постановке, подготовке и решению любого класса задач с применением компьютерных технологий.

Изучение дисциплины направлено на развитие логического мышления и навыков по составлению алгоритмов решения инженерных и прикладных задач с использованием алгоритмического языка Turbo Pascal 7.0.

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина «Информатика» относится к учебным дисциплинам базовой части профессионального цикла основной образовательной программы (далее - ООП) направления подготовки 190700 – «Технология транспортных средств», квалификация (степень) – бакалавр.

Для успешного освоения данной дисциплины студент должен владеть знаниями, умениями и навыками, сформированными школьной программой по дисциплине «Информатика и ИКТ», а также дисциплинами ООП бакалавриата: «Математика», «Физика», «Прикладная математика», в частности:

знать:

- основы компьютерной грамотности;
- основные понятия в Информатике;
- основы решения логических задач;

уметь:

- решать логические задачи;
- применять информационные технологии для решения прикладных задач;

владеть:

- навыками составления алгоритмических структур;
- навыками самостоятельной работы, самоорганизации и организации выполнения поручений;
- программами Microsoft Office для работы с деловой информацией.

Освоение дисциплины «Информатика», которая по учебному плану бакалавриата направления подготовки 190700 – «Технология транспортных средств», изучается на 1-ом курсе, необходимо для изучения технических дисциплин, включающих применение информационных и компьютерных технологий (например, «Информационные технологии в транспорте» и т.д.).

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

Процесс изучения дисциплины направлен на формирование у студентов компетенций:

общекультурных:

- владение культурой мышления, способность к обобщению, анализу, восприятию информации, постановке цели и выбору путей её достижения (ОК-1);
- умение логически верно, аргументированно и ясно строить устную и письменную речь (ОК-2);
- использование основных законов естественнонаучных дисциплин в профессиональной деятельности, применение методов математического анализа и моделирования, теоретического и экспериментального исследования (ОК-10);
- способность понимать сущность и значение информации в развитии современного информационного общества, осознавание опасности и угрозы, возникающие в этом процессе, соблюдать основные требования информационной безопасности, в том числе защиты государственной тайны (ОК-11);
- владение основными методами, способами и средствами получения, хранения, переработки информации, приобретение навыков работы с компьютером как средством управления информацией (ОК-12);
- способность работать с информацией в глобальных компьютерных сетях (ОК-13);
- способность приобретать новые знания, используя современные образовательные и информационные технологии (ОК-17).

профессиональных:

- готовность к разработке и внедрению технологических процессов, использованию технической документации, распорядительных актов предприятия (ПК-1);
- готовность применять новейшие технологии управления движением транспортных средств (ПК-14);
- способность использования современных информационных технологий как инструмент оптимизации процессов управления в транспортном комплексе (ПК-17);
- способность изучать и анализировать информацию, технические данные, показатели и результаты работы транспортных систем; использование возможностей современных информационно-компьютерных технологий при управлении перевозками в реальном режиме времени (ПК-25).

В результате изучения дисциплины "Информатика" студент должен:

знать:

- об аппаратном и программном обеспечении персональных компьютеров,
- о процедурах машинных вычислений,
- о методах эффективной работы на персональном компьютере,
- об алгоритмах решения математических и инженерных задач;
- основы программирования на алгоритмическом языке высокого уровня Turbo Pascal 7.0;

уметь:

- практически работать на персональном компьютере,
- эффективно использовать основные системные и прикладные программные средства,
- составлять алгоритмы решаемых прикладных задач,
- осуществлять реализацию прикладных программ на основе разработанных алгоритмов;

владеть:

- современными информационными технологиями,
- навыками разработки алгоритмов для решения прикладных и инженерных задач.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зачетных единиц, 180 часов.

№ п/п	Раздел дисциплины	Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Формы текущего контроля успеваемости (по неделям семестра)
			Л	Пр	Лаб	
1.	РАЗДЕЛ 1. ВВЕДЕНИЕ В ИНФОРМАТИКУ	2	2			
2.	Тема 1.1. Основы информатики		2			Самостоятельное изучение материала
3.	РАЗДЕЛ 2: АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ ОПЕРАЦИИ	2	6			
4.	Тема 2.1. Системы счисления		2			Проработка лекционного материала
5.	Тема 2.2. Операции над двоичными числами и над числами с плавающей точкой		2			КДЗ №1
6.	Тема 2.3. Основы алгебры логики. Аксиомы		2			Самостоятельное изучение материала
7.	РАЗДЕЛ 3. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА	2	4			
8.	Тема 3.1. Технические средства реализации информационных процессов		2			Проработка лекционного материала
9.	Тема 3.2. Периферийное оборудование		2			Самостоятельное изучение материала
10.	РАЗДЕЛ 4. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА	2	8		4	
11.	Тема 4.1. Программные средства реализации информационных процессов		2		4	Защита лаб. работы

12.	Тема 4.2. Прикладное программное обеспечение		4			Проработка лекционного материала
13.	Тема 4.3. Локальные и глобальные сети		2			Проработка лекционного материала
14.	РАЗДЕЛ. 5. АЛГОРИТМИЗАЦИЯ ПРИКЛАДНЫХ ЗАДАЧ	2	16	2	4	
15.	Тема 5.1. Понятие алгоритма. Свойства и формы представления алгоритмов		2			
16.	Тема 5.2. Реализация программ на алгоритмическом языке.		2	2	4	Защита лаб. работ
17.	Тема 5.3. Программное обеспечение и технологии программирования		2			КДЗ №2
18.	Тема 5.4. Типы данных. Форматы базовых данных. Работа со строковыми данными		2			
19.	Тема 5.5. Арифметические и логические операции. Операторы передачи управления. Операторы цикла		2			Проработка лекционного материала
20.	Тема 5.6. Массивы. Виды массивов. Операции с массивами		2			
21.	Тема 5.7. Модульное программирование. Процедуры и функции		2			Проработка лекционного материала
22.	Тема 5.8. Графика. Создание изображений в графическом режиме		2			Защита лаб. работ
23.	ПОДГОТОВКА К ЭКЗАМЕНУ					
24.	ИТОГО:		36	2	8	Форма аттестации - экзамен

Содержание дисциплины

РАЗДЕЛ 1. Введение в информатику

Лекция 1. Основы информатики (2 часа)

Введение. Цель, задачи и содержание дисциплины, ее связь с естественнонаучными и другими общепрофессиональными дисциплинами, прикладное значение в подготовке специалиста. Научные основы информатики.

Основные понятия, термины и определения. Общая характеристика процессов сбора, передачи, обработки и накопления информации. Технические и программные средства реализации информационных процессов.

Понятие сообщения, данных и сигналов. Единицы измерения, представления и хранения информации.

РАЗДЕЛ 2. Арифметические и логические операции (6 часов)

Лекция 2. Системы счисления (2 часа)

Позиционные и непозиционные системы счисления. Различия систем счисления. Запись чисел в позиционных системах счисления. Переводы из любой системы счисления в десятичную и обратно. Взаимосвязь между восьмеричной системой счисления и шестнадцатеричной. Выполнение арифметических вычислений в различных системах счисления.

Лекция 3. Операции над двоичными числами и над числами с плавающей точкой (2 часа)

Формы представления числа в памяти персонального компьютера. Способы размещения чисел с фиксированной и плавающей точкой. Выполнение арифметических операций над числами с фиксированной запятой в двоичной системе счисления, а именно: сложение, умножение, вычитание и деление. Приведены примеры алгебраического сложения с использованием обратного или дополнительного кода. Числа с плавающей точкой. Нормализация мантиссы. Операции над числами с плавающей точкой.

Лекция 4. Основы алгебры логики. Аксиомы (2 часа)

Основные понятия формальной логики. Высказывание, суждение. Истинность и ложность высказываний. Основные логические операции, формулы. Основные соотношения и аксиомы алгебры логики. Формы представления логических функций (ДНФ, КНФ, СДНФ, СКНФ). Преобразование логических формул. Минимизация логических функций аналитическим способом и методом карт Карно (диаграмм Вейча). Логические основы работы компьютера. Логические законы. Построение таблиц истинности с применением Excel. Построение схем с использованием логических элементов. Решение логических задач. Представление логических выражений в виде формул.

РАЗДЕЛ 3. Аппаратное обеспечение персонального компьютера (4 часа)

Лекция 5. Технические средства реализации информационных процессов (2 часа)

История развития электронно-вычислительных машин (ЭВМ), открытая архитектура ЭВМ. Принцип фон Неймана. Состав персонального компьютера,

назначение и характеристики основных элементов: центрального процессора, системных шин, системной памяти (ОЗУ, ПЗУ, КЭШ). Принцип работы вычислительных систем. Внешние и внутренние запоминающие устройства, устройства для хранения информации.

Лекция 6. Периферийное оборудование (2 часа)

Устройства ввода/вывода: основные типы, их назначение. Накопители на гибких, жестких дисках, на CD- и DVD-дисках (конструкции, характеристики). Клавиатура, состав конструкции, назначение клавиш.

Мониторы различных видов: ЭЛТ-, ЖКИ-, плазменные мониторы. Конструкции, характеристики, функционирование. Растровая и векторная графика. Цветовые модели: RGB, CMY, прочие.

Манипуляторы: мышь, джойстик, track-ball, track-point. Конструкции, функционирование. Сканеры, плоттеры, модемы. Конструкции, характеристики, функционирование.

Мобильные компьютеры: разновидности, характеристики.

РАЗДЕЛ 4. Программное обеспечение персонального компьютера (8 часов)

Лекция 7. Программные средства реализации информационных процессов (2 часа)

Роль программного обеспечения. Системное и прикладное программное обеспечение. Типы системных программ: операционные системы, программы-оболочки, операционные оболочки, программы-утилиты, драйверы. Операционные системы (ОС): назначение, классификация, состав, функции и характеристики. Версии, состав, команды операционных систем. Основные функции, выполнение основных операций. Состав, свойства, характеристики. Файловая система. Создание и удаление файлов. Смена каталога. Архивация файлов.

Программы-оболочки. Назначение, виды, функции. Norton Commander и его клоны (FAR, Total Comm., Windows Comm.). Программы-утилиты. Типы утилит. Проблемы борьбы с вирусами, типы вирусов, антивирусные программы: типы, свойства, использование (KAV, DrWeb, AidsTest, Norton Antivirus и пр.). Программы-архиваторы: типы, свойства, использование (ZIP, ARJ, RAR). Дисковые утилиты: проверка диска, дефрагментация диска.

Лекция 8-9. Прикладное программное обеспечение (4 часа)

Типы прикладного программного обеспечения. Табличный процессор EXCEL. Назначение, структура, основные функции. Способы ввода данных, формул и их редактирование. Типы данных. Ячейки, типы ссылок на ячейки, диапазон. Работа со списками. Диаграммы и графики. Макросы.

Назначение и основные функции текстовых процессоров (WINWORD, WORD). Приемы ввода, редактирования и форматирования текста. Установка параметров страницы. Колонтитулы, списки. Работа с таблицами. Вставка внешних объектов. Математические формулы. Расширенные возможности приложения. Гипертексты, ссылки в документе. Макросы.

Системы управления базами данных ACCESS. Принцип работы с объектами СУБД. Основные понятия реляционной модели данных. Проектирование баз данных, создание связи между таблицами. Обеспечение целостности данных. Заполнение таблиц баз данных. Построение запросов, формирование отчетов.

Электронные презентации. Этапы создания презентаций. Структура презентаций, назначение стилей. Создание слайда, добавление и удаление слайда. Анимация. Работа с разными режимами презентаций.

Графические редакторы. Способы представления и хранения графической информации. Форматы графических файлов. Операции с графическими объектами.

Языки программирования высокого уровня. Трансляторы–интерпретаторы и компиляторы. Служебные программы для систем программирования.

Лекция 10. Локальные и глобальные сети (2 часа)

Структуры информационных и информационно-вычислительных сетей: особенности, сравнительные характеристики. Основные требования к вычислительным сетям. Модели взаимодействия открытых сетей. Протоколы. Сеть INTERNET. Топология и архитектура сетей. Способы подключения компьютеров в сети. Принципы адресации. Удаленное управление компьютером. Телеконференция. Электронная почта. Службы передачи информации. Поисковые службы. Сетевые стандарты. Основы защиты информации и сведений в компьютерных сетях. Методы защиты информации: программные, аппаратные, организационные. Электронная подпись.

РАЗДЕЛ 5. Алгоритмизация прикладных задач (16 часов)

Лекция 11. Понятие алгоритма. Свойства и формы представления алгоритмов (2 часа)

Модели решения функциональных и вычислительных задач. Понятие алгоритма. Свойства алгоритмов. Типы и способы построения алгоритмов. Модели и процессы алгоритмизации и программирования прикладных инженерных задач. Машина Поста. Формы представления алгоритмов. Основные алгоритмические конструкции. Типовые алгоритмы. Использование типовых алгоритмов для решения задач.

Лекция 12. Реализация программ на алгоритмическом языке (2 часа)

Этапы создания программы. Надежность программы. Корректность программы. Тестирование и отладка программы. Проектирование тестов. Этапы тестирования. Методы отладки. Ошибки в программе. Способы обнаружения ошибок. Эффективность программы. Оценка сложности алгоритма.

Лекция 13. Программное обеспечение и технологии программирования (2 часа)

Понятие о методах программирования. Основные приемы. Требования к программам: дружелюбность пользователю, дружелюбность программисту,

управление работой компьютера. Стратегии разработки программ. Этапы разработки программ. Приемы программирования. Планирование и организация программ. Типовые процедуры. Обработка ошибок.

Лекция 14. Типы данных. Форматы базовых данных. Работа со строковыми данными (2 часа)

Типы данных. Базовые типы данных (булевы, целые, вещественные и символьные). Форматы базовых типов данных. Способы кодирования, операции и диапазоны значений базовых типов данных. Абстрактные типы данных: разновидности (строки, массивы, списки, очереди, записи). Построение абстрактных типов данных из базовых, особенности реализации.

Строковые операции. Конкатенация. Строковые процедуры и функции. Строковые функции модуля String. Способы реализации обработки строк. Примеры решения задач.

Лекция 15. Арифметические и логические операции. Операторы передачи управления. Операторы цикла (2 часа)

Арифметические операции. Логические операции. Операции отношения. Операторы передачи управления. Условный оператор IF. Сокращенная, полная и расширенная форма представления оператора IF. Оператор выбора Case. Логика работы операторов передачи управления. Операторы цикла: цикл со счетчиком, цикл с предусловием и цикл с постусловием. Логика работы. Примеры решения задач.

Лекция 16. Массивы. Виды массивов. Операции с массивами (2 часа)

Массивы. Виды, характеристики массивов. Задание массивов. Заполнение массивов. Операции с массивами. Поиск в массиве. Сортировка массива. Запись элементов массива в файл. Примеры решения задач.

Лекция 17. Модульное программирование. Процедуры и функции (2 часа)

Модули. Типовые процедуры. Структура процедур и функций. Область действия идентификаторов при использовании процедур и функций. Способы передачи параметров. Рекурсия. Формы рекурсивных процедур. Примеры решения задач.

Лекция 18. Графика. Создание изображений в графическом режиме (2 часа)

Предопределенные константы в графическом режиме. Предопределенные типы для установки цветовой палитры, вида линий, оформления текста. Инициализация графического режима. Ошибки инициализации. Система координат. Текущий указатель. Простейшие графические фигуры. Вывод текста в графическом режиме. Эффект движения в графическом режиме.

Перечень практических занятий и их объем в часах

№ п/п	Тема	Объем в часах
1.	Составление алгоритмов решения вычислительных задач. Составление программ вычислений. Составление программ с ветвлениями и циклами	2 часа
ИТОГО:		2 часа

Перечень лабораторных работ и их объем в часах

№ п/п	Тема	Объем в часах
1.	Расширенные возможности приложения Word. Изучение приемов работы в Excel.	4 часа
2.	Знакомство со средой языка Turbo Pascal. Составление простых вычислительных программ. Составление программ с ветвлениями.	4 часа
ИТОГО:		8 часов

5. Образовательные технологии

В процессе преподавания дисциплины «Информатика» используются классические формы и методы обучения (лекции, практические занятия и лабораторные работы, рефераты). Применение любой формы обучения предполагает также использование новейших IT-обучающих технологий.

При проведении лекционных занятий по дисциплине преподаватель использует компьютерные и мультимедийные средства обучения, а также раздаточные материалы.

Практические занятия проводятся с использованием компьютерного и мультимедийного оборудования, при необходимости - с привлечением Интернет-ресурсов.

Лабораторные работы выполняются в компьютерных аудиториях.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Текущий контроль успеваемости студентов по дисциплине «Информатика» включает отчет по практической работе, защиту лабораторных работ, выполнение индивидуальных контрольных домашних заданий.

Темы для подготовки контрольных домашних заданий:

№1. Позиционные и непозиционные системы счисления. Различия систем счисления. Запись чисел в позиционных системах счисления. Переводы из любой системы счисления в десятичную и обратно. Взаимосвязь между восьмеричной системой счисления и шестнадцатеричной. Выполнение арифметических вычислений в различных системах счисления. Способы размещения чисел

с фиксированной и плавающей точкой. Выполнение арифметических операций над числами с фиксированной запятой в двоичной системе счисления (сложение, умножение, вычитание и деление). Алгебраическое сложение с использованием обратного или дополнительного кода. Числа с плавающей точкой. Нормализация мантииссы. Операции над числами с плавающей точкой.

№2. Написание программы на алгоритмическом языке. Разработка алгоритма задачи. Этапы создания программы. Надежность программы. Корректность программы. Тестирование и отладка программы. Проектирование тестов. Этапы тестирования. Методы отладки. Ошибки в программе. Способы обнаружения ошибок.

Итоговая аттестация по итогам освоения дисциплины осуществляется в виде экзамена в соответствии с «Положением о балльно-рейтинговой системе контроля успеваемости студентов».

Вопросы для подготовки к экзамену

1. Понятия информации, сообщения, языка. Связь информации и сообщения.
2. Передача сообщений. Виды носителей, сигналы, знаки, алфавит.
3. Системы счисления (СС). Понятия цифр, чисел, основания, разряда, веса числа.
4. Правила перевода целых чисел из 10-ой СС в СС с иным основанием.
5. Правила перевода дробных чисел из 10-ой СС в СС с иным основанием.
6. Правила перевода чисел из СС с произвольным основанием в десятичную СС.
7. Правила перевода чисел из 8-ой СС в 16-ую СС и обратно.
8. Нормализация двоичных чисел.
9. Сложение и вычитание двоичных чисел с фиксированной запятой.
10. Умножение двоичных чисел с фиксированной запятой.
11. Алгебраическое сложение двоичных чисел в обратном коде.
12. Алгебраическое сложение двоичных чисел в дополнительном коде.
13. Сложение чисел с плавающей запятой.
14. Архитектура ЭВМ. Принципы фон Неймана.
15. Назначение, состав и функции центрального процессора.
16. Назначение, виды и особенности УВВ.
17. Типы накопителей для хранения информации.
18. Клавиатура: типы, назначение клавиш.
19. Принтеры: типы, особенности.
20. Манипуляторы.
21. Мобильные компьютеры.
22. Разновидности программ. Общая классификация системных программ.
23. Операционные системы: назначение, функции.
24. Программы–архиваторы.

25. Вирусы: типы, действия.
26. Антивирусы. Организационные методы борьбы с вирусами.
27. Редакторы текстов: типы, функции.
28. Табличные процессоры, СУБД – назначение, функции.
29. Графические редакторы – назначение, функции.
30. Инструментальные системы программирования.
31. Каталоги. Файлы.
32. Алфавит языка Turbo Pascal.
33. Структура программы в языке Turbo Pascal.
34. Операторы цикла.
35. Оператор FOR...
36. Функция DIV.
37. Функция MOD.
38. Определение четных чисел в языке Turbo Pascal.
39. Математические функции в Turbo Pascal.
40. Арифметические операции в Turbo Pascal.
41. Логические операции в языке Turbo Pascal.
42. Датчик случайных чисел.
43. Операторы для вывода данных.
44. Операторы передачи управления.
45. Условные операторы.
46. Оператор IF... THEN (строчный).
47. Оператор IF... ELSEIF (блочный).
48. Конкатенация.
49. Типы данных.
50. Массивы. Способы заполнения массивов.
51. Типы алгоритмических структур.
52. Формы представления алгоритмов.
53. Алгоритмы: суть, свойства.
54. Алгоритм. Графическое представление алгоритма.

Итоговая оценка по дисциплине определяется с учетом приведенной ниже шкалы.

Рейтинговая оценка	<50	50-59	60-69	70-79	80-89	90-100
5-ти бальная оценка	Неудовлетворительно	Удовлетворительно		Хорошо		Отлично
Европейская оценка	F	E	D	C	B	A

Самостоятельная работа студентов по дисциплине «Информатика» способствует более глубокому усвоению изучаемого курса, формирует навыки логического мышления и исследовательской работы по проблемам алгоритмизации задачи, ориентирует студента на умение применять полученные теоретические знания на практике и проводится в следующих видах:

- проработка лекционного материала;

- подготовка к выполнению и защите лабораторных работ;
- подготовка к практической работе;
- решение контрольных домашних заданий;
- подготовка к экзамену.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература

1. Основы современных компьютерных технологий /под ред. А.Д. Хомоненко. - СПб: Корона-принт, 1998.
2. Росс Г.В., Дулькин В.Н., Сысоева Л.А. Основы информатики и программирования. - М.: ПРИОР, 1999.
3. Аладьев В.З., Хунт Ю.Я., Шишаков М.Л. Основы информатики: учебное пособие. - М.: ИИД Филинь, 1998.
4. Симонович С.В. и др. Информатика. Базовый курс. - СПб: Питер, 2001.
5. Марченко А.И., Марченко Л.А. Программирование в среде Turbo Pascal 7.0 /под ред. В.П. Тарасенко. – СПб: Корона, 2004.
6. Моргун А.И. Справочник по Turbo Pascal для студентов. – М.: Изд. дом «Вильямс», 2006.
7. Культин Н.Б. Turbo Pascal в задачах и примерах. СПб: БХВ – Петербург, 2005.
8. Николаев А.Б., Акетнова Л.А. Турбо Паскаль в примерах: кн. для учащихся 10 – 11 кл. – М.: Просвещение, 2002.

б) дополнительная литература

9. Фигурнов В.Э. IBM PC для пользователя. - изд. 7-е, перераб. и доп. - М.: ИНФРА-М. 1997.
10. Левин А. Самоучитель полезных программ. - М.: Нолидж, 1999.
11. Лэнгсам Й., Огенстайн М., Тененбаум А. Структуры данных для персональных ЭВМ. - М.: Мир, 1989.
12. Лихтциндер Б.Я., Кузнецов В.Н. Микропроцессоры и вычислительные устройства в радиотехнике. – К.: Высшая школа, 1988.
13. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. – М.: МЦНМО, 1999.
14. Бауэр Ф.Л., Гооз Г. Информатика. Вводный курс. - М.: Мир, 1990. - Ч. 1,2.
15. Симпсон Г. Профессиональная работа на персональном компьютере. - М.: Финансы и статистика, 1998.
16. Новейший самоучитель работы на компьютере /под ред. С. Симоновича. М.: Десс; Инфорком-Пресс, 1999.
17. Основы экономической информатики /под ред. проф. А.Н. Морозевича. Минск: БГЭУ, 1998.
18. Пятибратов А.Л., Гуденко Л.Л., Кириченко А.А. Вычислительные системы, сети и телекоммуникации. - М.: «Финансы и статистика», 1998.
19. Экономическая информатика: учебник для вузов /под ред. проф. В.В. Евхимова. - СПб: Питер, 1997.

в) учебно-методическая литература по проведению практических занятий

20. Алексеев В.Е., Ваулин А.С., Петрова Г.Б. Вычислительная техника и программирование. Практикум по программированию: практическое пособие. - М.: Высшая школа, 1991.

21. Симонович С.В. и др. Информатика. Базовый курс. - СПб: Питер, 2001.

22. Культин Н.Б. Turbo Pascal в задачах и примерах. СПб: БХВ – Петербург, 2005.

г) учебно-методическая литература для выполнения лабораторных работ

23. Пегова Е.П. Методические указания по выполнению лабораторных работ по дисциплине «Информатика. Практикум, MS Word». - М.: Дрофа, 2008.

24. Пегова Е.П. Методические указания по выполнению лабораторных работ по дисциплине «Информатика. Практикум, MS Excel 2003». - М.: Дрофа, 2008.

25. Кишенский С.Ж., Невельская О.В. Методические указания по проведению лабораторных работ по дисциплине «Информатика». - М.: МГТУ ГА, 2010. – Ч. 1,2.

в) программное обеспечение и Интернет-ресурсы

✓ Электронные ресурсы библиотеки Университета, электронные версии пособий, методических разработок, указаний и рекомендаций по всем видам учебной работы.

✓

8. Материально-техническое обеспечение дисциплины

1. Учебный класс, оборудованный персональными компьютерами. На компьютерах необходимо наличие следующих программных приложений:

✓ **Microsoft Office Word**

✓ **Microsoft Office Excel**

✓ **Turbo Pascal 7.0**

Часть II. Методические указания по выполнению контрольных работ

Во второй части пособия излагаются теоретические положения, методические указания и варианты контрольных заданий. Выполнение задания заключается в письменном решении 6 задач по тематике "Вычисления в ЭВМ" (задания 1-6) и составлении программ на алгоритмическом языке высокого уровня для решения различных вычислительных и прикладных задач (задания 7-9).

При выполнении каждого задания следует придерживаться изложенных ниже требований:

а) при решении задач (задания 1-6) следует записать полный текст задачи с исходными данными в соответствии со своим вариантом. Решение задачи изложить подробно (запись процесса решения с промежуточными результатами);

б) при составлении программ (задания 7-9) следует составить алгоритмическую схему решения задачи со стандартными обозначениями блоков алгоритма и привести текст программы. Допускается использовать для реализации программы следующие языки высокого уровня: **Turbo Pascal, Бейсик, Фортран, Паскаль, Си** (примеры решения задач в данном пособии реализованы на языке **Turbo Pascal 7.0.**). После текста программы необходимо привести предполагаемое изображение на экране монитора диалога с компьютером при запуске программы, вводе исходных данных и выдаваемой на экран информации в процессе решения задачи. Допускается (при наличии доступа к компьютеру) использовать распечатки текстов программ и решений задач вместо ручной записи;

в) писать следует чернилами или шариковой ручкой. Чертежи (блок-схемы алгоритмов) выполнять с использованием стандартных обозначений. В тетради следует оставлять поля не менее 20 мм для заметок преподавателя. Страницы нумеровать обязательно.

1. Выбор вариантов заданий контрольной работы

Для всех заданий дано 30 вариантов задач. Номер варианта задачи одинаков для всех заданий и определяется числом, состоящим из двух последних цифр номера зачетной книжки, взятым по модулю числа 30, например:

- если номер зачетной книжки - 98878 (последние цифры - 78), то номер варианта: $78 \bmod 30 = 18$;
- если последние цифры зачетной книжки - 25, то номер варианта: $25 \bmod 30 = 25$.
- если последние цифры зачетной книжки - 00, то номер варианта: $100 \bmod 30 = 10$.
- если последние цифры зачетной книжки – 30, 60 или 90, то номер варианта - 30

Задание 1. Перевод чисел из десятичной системы счисления в систему счисления с основанием "b"

Теоретические положения.

Система счисления (СС) - совокупность символов и правил для записи чисел. СС разделяются на позиционные и непозиционные.

Хорошо знакомая нам десятичная система счисления является позиционной системой счисления и располагает только десятью цифрами (0,1,2,...,9), с помощью которых можно представить любое число. В *позиционной системе счисления* значение числа определяется местом расположения (позицией) каждой цифры в числе (321, 132).

Правила записи и выполнения различных операций во всех позиционных СС одинаковы, эти системы счисления отличаются друг от друга только *осно-*

ваем. Основание системы счисления - это количество цифр (символов), используемых для записи любого числа, так:

- в десятичной системе счисления используется десять цифр - $0 \div 9$;
- в двоичной системе счисления используются две цифры - $0 \div 1$;
- в восьмеричной системе - восемь цифр - $0 \div 7$;
- в шестнадцатеричной системе счисления задействовано шестнадцать символов - цифры $0 \div 9$ и буквы латинского алфавита A,B,C,D,E,F для записи чисел 10, 11, 12, 13, 14, 15 соответственно.

Пример 1. $75,35_{(10)}$ перевести в 2-ую системы счисления

Для получения частных и остатков по правилу деления для целой части числа удобно использовать формулу записи, известную под названием «деление в столбик», а для получения кода дробной части числа по правилу умножения «столбиком». Применительно к рассматриваемому примеру имеем:

В двоичную

0,	35								
	2								
0,	70								
	2								
1,	40								
	2								
0,	80								
	2								
1,	60								
	2								
1,	20								

75	2								
1	37	2							
	1	18	2						
	0	9	2						
	1	4	2						
	0	2	2						
	0	1	1						

В результате преобразования получаем $75,35_{(10)} = 1001011,01011..._{(2)}$.

Как следует из примера, процесс перевода дробной части можно продолжить до бесконечности. ЭВМ оперирует числами, представленными конечными наборами цифр. Поэтому дроби округляют в соответствии с правилами преобразования и весом младшего разряда исходной дроби.

Пример 2. $75,35_{(10)}$ перевести в 8-ую системы счисления

В восьмеричную

0,	35								
	8								
2,	80								
	8								
6,	40								
	8								
3,	20								
	8								
1,	60								
	8								
4,	80								

75	8								
3	9	8							
	1	1	1						

В результате преобразования получаем $75,35_{(10)} = 113,26314..._{(8)}$.

Пример 3. $75,35_{(10)}$ перевести в 16-ую системы счисления

В шестнадцатеричную

$$\begin{array}{r} 0 \overline{) 35} \\ \underline{16} \\ 5 \overline{) 60} \\ \underline{16} \\ 9 \overline{) 60} \\ \underline{16} \\ 9 \overline{) 60} \end{array} \quad \begin{array}{r} 75 \overline{) 16} \\ \underline{11} \\ 4 \end{array}$$

↘
B

В результате преобразования получаем $75,35_{(10)} = 4B,599..._{(16)}$.

Варианты задания № 1.

Перевести десятичное число в систему счисления с основанием "b".*

№	Число	b	№	Число	b	№	Число	b
1	37,46	2	11	1234,44	8	21	8213,73	16
2	79,18	2	12	2112,77	8	22	7231,52	16
3	66,38	2	13	1873,52	8	23	6972,73	16
4	59,85	2	14	2531,28	8	24	6331,68	16
5	77,72	2	15	1757,53	8	25	8649,39	16
6	66,74	2	16	2231,32	8	26	7453,27	16
7	84,56	2	17	2169,73	8	27	7979,74	16
8	67,36	2	18	1327,12	8	28	6397,78	16
9	87,44	2	19	1498,62	8	29	5685,62	16
10	66,4	2	20	2333,62	8	30	6377,34	16

Задание 2. Перевод чисел из системы счисления с основанием "b" в десятичную систему счисления

Теоретические положения.

Алгоритм перевода чисел из СС с основанием "b" в десятичную систему счисления заключается в выполнении следующих действий:

1. Записать исходное b-ое число в виде.
2. Все коэффициенты записать в десятичной системе счисления.

Пример 4: $101011,1011_2$ перевести в десятичную систему счисления

$$\begin{array}{cccccccc} 5 & 4 & 3 & 2 & 1 & 0 & -1 & -2 & -3 & -4 \\ 101011,1011_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ + 1 \cdot 2^{-4} = 32 + 0 + 8 + 0 + 2 + 1 + 0,5 + 0 + 0,125 + 0,0625 = 43,687510 \\ 101011,1011_2 = 43,6875_{10}. \end{array}$$

* Ход вычислений должен быть приведен полностью (аналогично примерам 1-3).

Пример 5: $123467,1011_8$ перевести в десятичную систему счисления

$5 \ 4 \ 3 \ 2 \ 1 \ 0 \ -1 \ -2 \ -3 \ -4$

$$123467,1011_8 = 1 \cdot 8^5 + 2 \cdot 8^4 + 3 \cdot 8^3 + 4 \cdot 8^2 + 6 \cdot 8^1 + 7 \cdot 8^0 + 1 \cdot 8^{-1} + 0 \cdot 8^{-2} + 1 \cdot 8^{-3} + 1 \cdot 8^{-4} = 32768 + 2 \cdot 4096 + 3 \cdot 512 + 4 \cdot 64 + 7 \cdot 1 + 1/8 + 0 + 1/512 + 1/4096 = 32768 + 8192 + 1536 + 256 + 8 + 585/4096 = 42760 + 0,127197 = 42760,127197_{10}$$

$$123467,1011_8 = 42760,127197_{10}$$

Пример 6: $10A1CF,1011_{16}$ перевести в десятичную систему счисления

$5 \ 4 \ 3 \ 2 \ 1 \ 0 \ -1 \ -2 \ -3 \ -4$

$$10A1CF,1011_{16} = 1 \cdot 16^5 + 0 \cdot 16^4 + 10 \cdot 16^3 + 1 \cdot 16^2 + 12 \cdot 16^1 + 14 \cdot 16^0 + 1 \cdot 16^{-1} + 0 \cdot 16^{-2} + 1 \cdot 16^{-3} + 1 \cdot 16^{-4} = 1048576 + 0 + 40960 + 256 + 192 + 14 + 1/16 + 0 + 1/4096 + 1/65536 = 1089998 + 4113/65536 = 1089998,06276_{10}$$

$$10A1CF,1011_{16} = 1089998,06276_{10}$$

Варианты задания 2.

Перевести число с основанием "b" в десятичную СС*

№	b	Число	№	b	Число	№	b	Число
1	2	101111,111	11	8	2551,72	21	16	4C9,A3
2	2	101011,110	12	8	6422,64	22	16	A3F,1C
3	2	110111,001	13	8	5262,35	23	16	98D,7E
4	2	101001,101	14	8	4341,63	24	16	B65,C6
5	2	101100,010	15	8	4364,32	25	16	459,E9
6	2	100111,110	16	8	4166,64	26	16	A86,AA
7	2	110010,111	17	8	2761,17	27	16	D49,A6
8	2	111010,011	18	8	6651,45	28	16	6EA,B5
9	2	101100,100	19	8	6523,22	29	16	C68,95
10	2	110111,111	20	8	4526,55	30	16	BCA,16

Задание 3. Перевод чисел из СС с основанием 8 в СС с основанием 16 (или обратно)

Преобразование чисел из восьмеричной системы счисления в шестнадцатеричную и обратно осуществляется через двоичную систему счисления. Основания этих систем счисления кратны целой степени 2, т.е. $8=2^3$, а $16=2^4$. Это означает, что при преобразовании восьмеричного кода числа в двоичный, необходимо каждую восьмеричную цифру заменить соответствующим трехзначным двоичным кодом (триадой).

* Процесс решения должен быть представлен аналогично примерам 4 - 6.

При преобразовании шестнадцатеричного кода числа в двоичный необходимо каждую шестнадцатеричную цифру заменить четырехзначным двоичным кодом (тетрадой).

При преобразовании двоичного кода в восьмеричный или шестнадцатеричный двоичный код делится соответственно на триады или тетрады влево и вправо от запятой (точки), разделяющей целую и дробные части числа. Затем триады (тетрады) заменяются восьмеричными (шестнадцатеричными) цифрами.

Если при разбиении двоичного кода в крайних триадах (тетрадах) недостает цифр до нужного количества, они дополняются нулями. Соответственно, «лишние» нули слева и справа, не вошедшие в триады (тетрады), отбрасываются.

Таблица взаимосвязи 2-ой и 8-ой системами счисления

8-ая с/с	0	1	2	3	4	5	6	7
2-ая с/с	000	001	010	011	100	101	110	111

Таблица взаимосвязи 2-ой и 16-ой системами счисления

16-ая с/с	0	1	2	3	4	5	6	7
2-ая с/с	0000	0001	0010	0011	0100	0101	0110	0111
16-ая с/с	8	9	10 (A)	11 (B)	12 (C)	13 (D)	14 (E)	15 (F)
2-ая с/с	1000	1001	1010	1011	1100	1101	1110	1111

Пример 7: 251, 56₍₈₎ преобразовать в 16-ую систему счисления

251, 56₍₈₎ = 010 101 001, 101 110₍₂₎ = 0000 1010 1001, 1011 1000₍₂₎ = 0A9, B8₍₁₆₎.

Пример 8: A3FD1,15DE₍₁₆₎ преобразовать в 8-ую систему счисления

A3FD1,15DE₍₁₆₎ = 1010 0011 1111 1101 0001, 0001 0101 1101 1110₍₂₎ = 010 100 011 111 111 010 001, 000 101 011 101 111 000₍₂₎ = 2437721, 053570₍₈₎.

Варианты задания 3.

Перевести 8-е (16-ричное) число в 16-ричную (8-ричную) СС*.

№	8 → 16	№	8 → 16	№	16 → 8	№	16 → 8
1	764,453	9	625,716	16	87C, 6A9	24	DF2, 7EC
2	431,532	10	231,673	17	938, DA1	25	89C, 94D
3	175,367	11	471, 573	18	95D, 4F8	26	874, CDF
4	126,761	12	726, 414	19	793, 6EC	27	D51, 963

* При переводе чисел из одной СС в другую следует пользоваться правилами, изложенными выше и представленными в примерах 1-6.

5	375,641	13	346, 573	20	68F, 79C	28	1CD, 77A
6	672,517	14	736, 241	21	BC4, E58	29	AC4, 6F5
7	264,375	15	616, 472	22	147, 984	30	651, 83D
8	517,264			23	16C, 67A		

Задание 4. Арифметические операции в двоичной системе счисления

Сложение

Правила сложения двоичных чисел аналогичны правилам в десятичной системе счисления.

Таблица сложения в двоичной системе счисления:

$$1 + 1 = 10 \qquad 1 + 0 = 1 \qquad 0 + 1 = 1 \qquad 0 + 0 = 0$$

Здесь все числа двоичные.

Пример 9: Складывать будем «столбиком» как и в десятичной системе счисления.

Сложим числа 1001 и 10

$$\begin{array}{r} 1001 \\ + 10 \\ \hline 1011 \end{array}$$

Сложим 100 и 100

$$\begin{array}{r} 100 \\ +100 \\ \hline 1000 \end{array}$$

Сложим 101 и 101

$$\begin{array}{r} 101 \\ +101 \\ \hline 1010 \end{array}$$

Вычитание

Таблица вычитания двоичных чисел:

$$10 - 1 = 1 \qquad 1 - 0 = 1 \qquad 1 - 1 = 0 \qquad 0 - 0 = 0$$

Пример 10: Вычитать будем «столбиком», как и в десятичной системе счисления.

Вычтем
из числа 100 число 10

$$\begin{array}{r} 100 \\ -10 \\ \hline 10 \end{array}$$

Вычтем
из числа 1001 число 10

$$\begin{array}{r} 1001 \\ -10 \\ \hline 111 \end{array}$$

Вычтем
из числа 1000 число

$$\begin{array}{r} 1 \\ 1000 \\ -1 \\ \hline 111 \end{array}$$

Здесь сначала из правого нуля в 100 вычли ноль, а затем, чтобы из среднего нуля в 100 вычесть 1, мы заняли 1 из позиции 3. Всё как в десятичной системе, но в нашем распоряжении только 0 и 1 и таблица вычитания.

Умножение

Таблица умножения двоичных чисел:

$$1 \cdot 1 = 1 \qquad 1 \cdot 0 = 0 \qquad 0 \cdot 1 = 0 \qquad 0 \cdot 0 = 0$$

Пример 11: Умножать будем «столбиком» как и в десятичной системе счисления.

Умножим числа 100 и 11

$$\begin{array}{r} 100 \\ *11 \\ \hline 100 \\ +100 \\ \hline 1100 \end{array}$$

Умножим числа 100 101

$$\begin{array}{r} 100 \\ *101 \\ \hline 100 \\ +100 \\ \hline 10100 \end{array}$$

Деление

Таблица деления двоичных чисел:

$1 : 1 = 1$

$0 : 1 = 0$

$0 : 0 = 0$

При делении используем таблицы вычитания и умножения для двоичных чисел, делим подобно тому, как мы делим десятичные числа.

Пример 12: разделим двоичные числа 1110 и 10

$$\begin{array}{r}
 1110 \overline{)10} \\
 \underline{-10} \quad 111 \\
 \quad 11 \\
 \quad \underline{-10} \\
 \quad \quad 10 \\
 \quad \quad \underline{-10} \\
 \quad \quad \quad 0
 \end{array}$$

Варианты задания 4.

Выполнить арифметические операции над числами в 2-ой СС

№	Арифметическая операция	№	Арифметическая операция	№	Арифметическая операция
1	1011+1111	11	10110·1011	21	4C9,A3
2	11001-10	12	101011+10101	22	A3F,1C
3	110101+111011	13	11110·110	23	110110-1011
4	1010100-1111	14	11111-1011	24	10101·10111
5	10111·11110	15	10111·101	25	11101·11011
6	11011·10101	16	10101-111	26	1101101+11001
7	100101+110111	17	11011·10100	27	1011011·101011
8	1001101-10011	18	1011·110	28	100010-111
9	1101011·10101	19	10000-11	29	110110·110
10	110110+11111	20	100100-1101	30	1101101+110111

Задание 5. Сложение целых двоичных чисел в обратном или дополнителном коде

Теоретические положения.

Представление чисел в компьютере по сравнению с формами, известными всем со школы, имеет два важных отличия:

- во-первых, числа записываются в двоичной системе счисления (в отличие от привычной десятичной);
- во-вторых, для записи и обработки чисел отводится конечное количество разрядов (в "некомпьютерной" арифметике такое ограничение отсутствует).

Двоичная арифметика предельно проста:

<p>Сложение производится согласно таблице сложения, которая для двоичных чисел имеет вид:</p> $0 + 0 = 0$ $0 + 1 = 1$ $1 + 0 = 1$ $1 + 1 = 10$	<p>Умножение производится согласно таблице умножения, которая для двоичных чисел имеет предельно простой вид:</p> $0 \cdot 0 = 0$ $0 \cdot 1 = 0$ $1 \cdot 0 = 0$ $1 \cdot 1 = 1$
--	---

В компьютерах арифметические устройства выполняют действия не с самими двоичными числами по правилам двоичной арифметики, а с их двоичными кодами (представлениями) по правилам арифметики двоичных кодов.

Отличия правил арифметики двоичных кодов от правил обычной арифметики заключаются в **ограниченности разрядной сетки**, применяемой для записи чисел в компьютере. Иначе говоря, для записи числа в устройствах компьютера выделяется фиксированное количество двоичных разрядов. Память компьютера имеет байтовую структуру, однако, размер одной адресуемой ячейки обычно составляет несколько байт: 2, 4, 8 байт.

Вся информация в ЭВМ представляется в двоичных кодах (**прямой, обратный и дополнительный**).

Для записи целого двоичного числа в **прямом** коде двоичные числа дополняются знаковым разрядом, который принимается равным "0" для положительных чисел и "1"- для отрицательных. При ручной записи чисел со знаком - знаковый разряд для удобства отделяется от значащих разрядов точкой. Например, десятичное число (+12) в прямом коде запишется так: (0.1100), а десятичное число (-12) – (1.1100).

Другими формами представления чисел со знаком являются **обратный и дополнительный** коды. Эти коды позволяют заменить вычитание чисел их сложением (исходя из принципа: $a - b = a + (-b)$).

Положительные числа, записанные в прямом, обратном и дополнительном кодах одинаковы.

Положительное десятичное число 12 в прямом, обратном и дополнительном двоичном кодах запишется: 1.1100.

Для перевода отрицательного числа из прямого кода в обратный следует в знаковом разряде сохранить единицу, а цифры значащих разрядов инвертировать, т.е. "1" заменить на "0", а "0" на "1".

Дополнительный код отрицательного числа получается из обратного кода числа прибавлением "1" к младшему разряду этого числа.

Пример 13: Записать десятичное число (-12) в прямом, обратном и дополнительном двоичном кодах в шестиразрядной ячейке:

1.01100 - прямой	1.10011 - обратный код	1.10100 - дополнит. код
------------------	------------------------	-------------------------

В данном примере один разряд отведен под знак числа, пять разрядов под само число. Само число сдвинуто к правому краю, а в избыточный разряд запи-

сан «0» в прямом коде. Затем прямой код инвертируется для перевода в обратный (под точку в разрядной сетке место не выделяется).

Перевод чисел из обратного (дополнительного) кода в прямой код производится по тем же правилам, что и в обратный (дополнительный) код из прямого.

Правила сложения в дополнительном коде:

1. Сложение производится по правилам сложения двоичных чисел, включая знаковый разряд ($0+0=0$, $0+1=1+0=1$, $1+1=10$ - то есть, "0" в данном разряде и перенос "1" в следующий по старшинству двоичный разряд).


2. Если в результате сложения возникает перенос (переполнение) из знакового разряда, этот перенос **игнорируется** (отбрасывается).

3. Если знак суммы не совпадает со знаками слагаемых (эта ситуация может возникнуть только когда они одинаковы), имеет место переполнение разрядной сетки ЭВМ и результат должен быть признан неверным.

Сложение в **обратном** двоичном коде отличается от сложения в **дополнительном** коде лишь одним правилом:

если в результате сложения возник перенос из знакового разряда (переполнение), **необходимо к младшему разряду суммы прибавить "1"**.

Пример14: Реализовать операцию 15-7 в прямом, обратном и дополнительном коде:

	10-е число	Прямой код	Обр. код	Доп. код
данные	15 - 7	0.1111 1.0111	0.1111 +1.1000	0.1111 +1.1001
промежуточный рез-т	8		10.0111 +  1	=10.1000
окончательный рез-т	8		0.1000	0.1000

Пример 15: Реализовать операцию: 7-15 в прямом, обратном и дополнительном коде:

	10-е число	Прямой код	Обр. код	Доп. код
данные	-15 + 7	1.1111 0.0111	1.0000 +0.0111	1.0001 +0.0111
промежуточный рез-т	- 8		1.0111	1.1000
окончательный рез-т (в прямом коде)	- 8		1.1000	1.0111 <u>+ 1</u> 1.1000

Варианты задания 5.

Осуществить алгебраическое сложение целых двоичных чисел в обратном (дополнительном) коде. Результат представить в прямом коде. *

№	a	b	код	№	a	b	код	№	a	b	код
1	16	-29	обр.	11	19	-24	обр.	21	26	-29	обр.
2	-24	17	доп.	12	-16	29	доп.	22	23	-18	доп.
3	29	-15	обр.	13	25	-27	обр.	23	29	-24	обр.
4	-21	24	доп.	14	-14	21	доп.	24	26	-26	доп.
5	25	-28	доп.	15	28	-25	доп.	25	25	-18	доп.
6	-28	23	обр.	16	-23	19	обр.	26	28	-26	обр.
7	26	-21	доп.	17	21	-26	доп.	27	28	-14	доп.
8	-24	19	обр.	18	-29	24	обр.	28	21	-26	обр.
9	24	-25	обр.	19	25	-27	обр.	29	24	-26	обр.
10	-26	19	доп.	20	29	-24	доп.	30	16	-23	доп.

Задание 6. Сложение чисел с плавающей запятой

Теоретические положения.

При представлении чисел с плавающей запятой (в показательной форме) числа имеют вид правильной дроби:

$$N = a \cdot b^p,$$

где a - мантисса (она является правильной дробью со знаком);

b - основание системы счисления (в ЭВМ лишь подразумевается);

p - порядок (целое число со знаком).

Десятичное число 23,73 может быть записано в следующих вариантах:

$$\begin{array}{lll} 1) 23,73 \cdot 10^0, & 2) 2,373 \cdot 10^1, & 3) 0,2373 \cdot 10^2, \\ 4) 0,02373 \cdot 10^3, & 5) 23730 \cdot 10^{-3} & \text{и т.д.} \end{array}$$

Чтобы исключить неоднозначность записи, используют *нормализованную* форму записи чисел с плавающей запятой. В этой форме мантисса числа имеет нулевую целую часть, а в старшем разряде дробной части - цифру, отличную от нуля (для двоичной системы - всегда "1").

Аналогично представляются числа с плавающей запятой и в двоичной системе.

* Процесс решения предполагает перевод десятичных данных в требуемый двоичный код и выполнение операции сложения аналогично примерам в задании 4.

Двоичное число $101,011$ в нормализованной показательной форме имеет вид: $0,101011 \cdot 10^{11}$.

Здесь основание "10" - запись десятичного числа "2" в двоичной системе счисления, а показатель "11" - двоичный аналог десятичного числа "3", компенсирующий сдвиг мантииссы на три разряда вправо при получении нормализованной формы.

Сложение чисел с плавающей запятой осуществляется в соответствии со следующим алгоритмом:

1. Уравнять порядки слагаемых. Для этого меньший порядок увеличивается до большего; при этом соответственно сдвигается мантиисса корректируемого числа. Так как число разрядов мантииссы (как и порядка) постоянно и задано разрядной сеткой ЭВМ, младшие разряды преобразуемого числа, выходящие за пределы разрядной сетки, теряются.

2. Выполняется (для отрицательных чисел) преобразование мантиисс в обратный (дополнительный) код.

3. Производится суммирование мантиисс по правилам алгебраического сложения двоичных чисел.

4. К сумме приписывается порядок слагаемых.

5. В случае переполнения производится нормализация результата (сдвиг мантииссы до получения нормализованной формы с соответствующим изменением значения порядка).

Выравниваются порядки слагаемых: *младший порядок увеличивается до большего*. При этом на соответствующее число разрядов *вправо* сдвигается мантиисса выравниваемого слагаемого. Производится суммирование мантиисс по правилам сложения чисел с фиксированной точкой в двоичном коде. К полученному результату добавляется порядок слагаемых. При необходимости производится *нормализация* мантииссы.

Пример 16: сложить 2 числа $0,10101 \cdot 10^{101}$ и $0,11001 \cdot 10^{011}$

Наименьший порядок у 2-го слагаемого: $0,11001 \cdot 10^{011} = 0,0011001 \cdot 10^{101}$

Производим сложение $0,10101$ и $0,0011001$:

$$\begin{array}{r} 0,10101 \\ + 0,0011001 \\ \hline 0,1101101 \end{array}$$

Получаем результат $0,1101101 \cdot 10^{101}$

В полученном результате мантиисса нормализована.

Пример 17: сложить 2 числа $0,11101 \cdot 10^{101}$ и $0,11001 \cdot 10^{011}$

Наименьший порядок у 2-го слагаемого: $0,11001 \cdot 10^{011} = 0,0011001 \cdot 10^{101}$

Производим сложение $0,11101$ и $0,0011001$:

$$\begin{array}{r} 0,11101 \\ + 0,0011001 \\ \hline 1,0001101 \end{array}$$

Получаем результат $1,0001101 \cdot 10^{101}$

В полученном результате мантисса сформировалась ненормализованной.
 Нормализация результата: $1,0001101 \cdot 10^{101} = 0,10001101 \cdot 10^{110}$

Варианты задания 6

Сложить два двоичных числа в форме с плавающей запятой. В ответе должны быть приведены все промежуточные результаты реализации алгоритма сложения.

№	1-е число		2-е число		№	1-е число		2-е число	
	мант	пор	мант	пор		мант	пор	мант	пор
1	0,10111	101	0,10100	010	16	0,11001	100	0,11101	010
2	0,11111	101	0,10101	011	17	0,11101	101	0,11011	011
3	0,10111	101	0,10111	011	18	0,11101	100	0,10110	101
4	0,11101	011	0,10101	100	19	0,10101	011	0,11001	100
5	0,10001	101	0,11101	011	20	0,10111	100	0,10111	011
6	0,11101	011	0,11010	100	21	0,11111	010	0,11101	100
7	0,11010	100	0,10110	101	22	0,10011	101	0,11110	011
8	0,11110	011	0,11011	110	23	0,10110	011	0,10010	101
9	0,10111	110	0,11111	100	24	0,11110	010	0,11011	100
10	0,11101	100	0,10010	101	25	0,10111	101	0,11100	100
11	0,11011	011	0,10111	100	26	0,11011	011	0,11011	100
12	0,11011	011	0,10001	101	27	0,10010	011	0,11011	110
13	0,11001	010	0,11100	011	28	0,11011	011	0,11001	100
14	0,10101	110	0,10111	100	29	0,11110	101	0,10111	100
15	0,11100	011	0,11011	100	30	0,11001	011	0,11110	101

Задание 7. Программирование простых вычислений

Теоретические положения.

Алгоритм - это последовательность действий, однозначно определяющих процесс преобразования исходных и промежуточных данных в результат решения задачи. Форма представления алгоритма может быть как текстовой, так и графической - в виде схемы. Решение всего многообразия задач может

быть сведено к трем типам алгоритмов: *линейному, разветвляющемуся и циклическому*. Чаще встречается комбинация этих типов.

Линейный алгоритм - алгоритм, в котором к результату решения задачи приводит *последовательное* выполнение действий.

Алгоритм разветвляющейся структуры - алгоритм, в котором последовательность выполнения действий *зависит от каких-либо условий*.

Циклический алгоритм - алгоритм, в котором для получения результата требуется *многократное (циклическое) повторение* ряда действий.

Программа на языке Turbo Pascal - это последовательность строк, описывающих алгоритм решения задачи. Строка может содержать один или несколько операторов, разделенных точкой с запятой ";".

Структура программы Turbo Pascal

Program [имя программы]

Uses [имя модуля]

Label [описание меток]

Const [описание констант]

Type [описание типов]

Var [описание переменных]

Procedure [имя процедур]

Function [имя функций]

Exports [описание экспортируемых имен]

Begin

[тело программы]

end.

Оператор представляет собой строго формализованное указание на выполнение конкретного действия. Длина программной строки не должна превышать 256 символов.

Данные представляются в программе в виде констант и переменных. Тип данных определяет возможные значения констант и переменных, форму представления в ЭВМ, объем занимаемой памяти, операции, которые могут выполняться над данными этого типа.

Числа. В языке *Turbo Pascal* пользуются двумя типами чисел: вещественными и целыми.

Под целое число отводится 1 байт или 2 байта памяти и оно хранится в форме с фиксированной точкой. Запись целого числа представляет собой последовательность цифр со знаком или без него (например: 120, -13, 5487, -7821, +3841).

Вещественные числа хранятся в ячейке памяти длиной 4 байта в форме с плавающей точкой. Возможны две формы "внешней" записи вещественных чисел в программах:

- с фиксированной точкой (например, - 3.7);
- с плавающей точкой (например, -0.45E2, 0.78E-3, здесь буква «Е» означает основание «10» и разделяет мантиссу и порядок).

Числовое или символьное значение можно сохранить в переменной или в константе.

Переменная - величина, которая может меняться при выполнении программы. Переменная всегда имеет имя, которое содержит не более 40 буквенно-цифровых символов и начинается с латинской буквы. Переменные описываются явно с помощью операторов описания типа (табл. 1).

Таблица 1

Таблица диапазона числовых данных

тип	диапазон
целый	-32768 ÷ +32767
целочисленный короткий	-128 ÷ +127
целочисленный длинный	-2147483648 ÷ +2147483647
веществ. обычной точности	-3.402823E+38 ÷ -1.40129E-45 +1.40129E-45 ÷ +3.402823E+38
веществ. двойной точности	-1.79769E+308 ÷ -4.94965E-324 +-4.94965E-324 ÷ +1.79769E+308

Программирование линейных алгоритмов

Программы, реализующие линейный алгоритм, как правило, очень просты и для их реализации используются операторы ввода данных, выполнения действий (вычислений) и вывода результатов.

Ввод данных в программах, написанных на языке *Turbo Pascal*, можно осуществить с помощью оператора ***Read, Readln***:

Read (a, b, c), Readln (a, b, c) - операторы ввода данных в диалоговом режиме, где a, b, c - список имен переменных.

Задать исходные данные можно также операторами присваивания, например: ***a := 5 ; b := -8 ; c := 12.5;***

Для вычисления арифметических выражений используется оператор **присваивания** «:=», частный случай которого может применяться и для ввода данных.

Общий вид оператора: ***Q := P***, где ***Q*** - имя переменной, ***P*** - арифметическое выражение.

Арифметические выражения соответствуют общепринятым алгебраическим выражениям, в них могут входить числа, переменные, функции, соединенные знаками арифметических операций и круглыми скобками.

В языке Turbo Pascal имеются следующие арифметические операции:

- *, /*** - умножение, деление;
- MOD*** - определение остатка от деления;
- DIV*** - целочисленное деление;
- +, -*** - сложение, вычитание.

Операции перечислены в порядке убывания приоритета их выполнения. Действия внутри круглых скобок выполняются первыми.

Математические функции

При записи функций на языке Turbo Pascal *аргумент функции* заключается в круглые скобки. В качестве аргумента математических функций может быть число, переменная или арифметическое выражение. Следует отметить, что в тригонометрических функциях аргумент должен быть задан в радианах.

Наиболее распространенные функции языка **Turbo Pascal**:

Abs(X) - вычисляет модуль аргумента, что соответствует математической записи $|x|$;

Exp(X) - экспонента, соответствует математической записи e^x ,

Ln(X) - вычисляет натуральный логарифм аргумента;

Sqr(x) - вычисляет возведение аргумента в квадратную степень;

Sqrt(X) - вычисляет корень квадратный из аргумента;

Cos(X) - вычисляет косинус аргумента;

Sin(X) - вычисляет синус аргумента;

Sgn(X) - определяет знак аргумента. Если аргумент отрицательный, функция принимает значение (-1,) если положительный (+1). При нулевом аргументе функция принимает значение 0.

Fix(X) - отбрасывает дробную часть значения аргумента;

Int(X) - округляет аргумент в сторону уменьшения;

Cint(X) - округляет аргумент по математическим правилам.

Random(X) - выдает случайное число обычной точности в интервале $0 \div 1$. Аргумент может быть опущен. Рекомендуется в начале программы запустить генератор случайных чисел оператором **Randomize**.

Write(x, y, z); WriteLn(x, y, z); - оператор вывода значений переменных **x, y, z**.

Пример18. Составить программу для вычисления формулы

Program zadacha;

Var

rez, a, b, x: real;

Begin

writeln('введите значения a, b, x');

readln(a, b, x);

rez:= x+(a/(7*b*x+a*x+cos(3*x)))-sin(x);

writeln('Результат вычислений =',rez);

end.

Операторы программы:

- **WriteLn** и **ReadLn** используются для организации диалога пользователя с компьютером.

- **-WriteLn** предназначен для вывода на экран текста, заключенного в апострофы, или значений переменных.

$$x + \frac{a}{x + 7bx + \cos 3x} - \sin x$$

- **Readln** служит для ввода числовых значений переменных.
- оператор присваивания имеет вид "**X:=Y**". Слева от знака присваивания " := " переменная (X), справа арифметическое выражение (Y). После вычисления (Y) полученное значение присваивается переменной (X).
- **END** окончание программы.

Пример 19: Преобразовать время в минутах в часы и минуты

Program zadacha_7;

Var

min, h, m : integer;

Begin

writeln('введите любое число в минутах');

readln(min);

h:= minDiv60;

m := min-h*60;

writeln (min, ' Составляет', h, ' час.', m, ' мин. ');

end.

Варианты задания 7

Составить программу вычисления и печати значений следующих выражений для исходных данных (данные подобрать самостоятельно), вводимых с клавиатуры ПК в диалоговом режиме:

1. $6y \cdot 2x + 4^y + \log_3(7x^2)$
2. $7,45e^{xy} + 6,98 \sin(xy)$
3. $a^x \cdot [\ln(x)]^y \cdot \cos[(1 - x^3/y)]$
4. $\sin(|y|/x) + y^2 - x^2 + 1$
5. $\sin(y) - y^{-1} + |\cos(x)| + x$
6. $x + a / [7bx^2 - 4/ax + \cos(3/4x)] - \sin(x)$
7. $x^3 + 1/y + \log_3(7x)$
8. $xy / (x+y) \cdot \sin[(1 - x^2/y^2)]$
9. $|(x+1)^2 - y^3| \cdot \sin(x)$
10. $6y \cdot 2x + 4xy - \sin(x - 2y) / \cos b \cdot (2x - y)$
11. $8x^2 - 2y^3 + \cos(x/y)$
12. $\log_3 y + \log_4 x^2 + \log_5(xy) + \ln(x)$
13. $\sin(|x + y|) + \cos(xy)$
14. $|2x - y| / \ln(x + y) + |\sin(y)|$
15. $(3x + yx - 6)^2 \cdot \ln(x + y) \cdot \sin y$
16. $(3x + 1)^2 \cdot [y + \cos(x)]^2 + \ln(x + y) / \ln(x - y)$
17. $|x| + y [74x(y - 9x)^2]^2 \cdot (\sin x)^2$
18. $[y - \sin(x)] / \log_9(xy) + (x - 2y)^2$
19. $\ln x + \log_3(7x^2) - |\sin(x+y) / \cos(x-y)|$
20. $x - 10 \cdot 2x - \sin(3x / (x - 1))$
21. $\cos(x) + 4y + \log_3(7x^2)$
22. $4x + \sin(y) - y^2 + |\cos(x)|$

23. $y^2 + x^2 + 5xy + 78$
24. $2\sqrt{\frac{\sin(x+y)}{x(x+1)} + \frac{x - \cos(xy)}{7+y}}$
25. $(1/x)^2 + (1/y)^2 + \cos(xy)$
26. $[2 \sin(x) - \sin(x)]$
27. $2\cos x + 4e^x + [ax + b(zxc)]^2 \div$
28. $3,22y + 6,27\sin(x)$
29. $5x + 2\sin(y) - 4/(x+7,2) + abc^2$
30. $6,28x + 3,14y \cdot \ln\left(\frac{1-x}{1+x}\right)$

Задание 8. Программирование ветвящихся алгоритмов

Теоретические положения.

Алгоритм разветвляющейся структуры - алгоритм, в котором последовательность выполнения действий зависит от каких-либо условий.

В языке *Turbo Pascal* для ветвления используются следующие операторы:

1. Оператор безусловной передачи управления **GOTO N**, где N метка строки. Этот оператор передает управление строке с меткой N.

2. Операторы условной передачи управления (приведены три типа):

а) **IF** <логическое выражение> **THEN** <оператор>;

- при выполнении оператора IF сначала определяется результат логического выражения: ИСТИНА (TRUE) или ЛОЖЬ (FALSE). Если ИСТИНА, то управление передается операторам, следующим за словом THEN, если - ЛОЖЬ, то оператору, записанному после оператора IF;

б) **IF** <логическое выражение> **THEN** <оператор> **ELSE** <оператор>;

- при выполнении оператора IF данной модификации сначала определяется результат логического выражения. Если ИСТИНА, то управление передается операторам, следующим за словом THEN, если - ЛОЖЬ, то оператору записанному после ELSE;

в) блочный **IF** (записывается в нескольких строках):

IF <логическое выражение> **THEN** <оператор>

[ELSE IF <логическое выражение> **THEN** <оператор>

ELSE <оператор> **]***;

При выполнении блочного IF, сначала определяется результат первого логического выражения. Если результат - ИСТИНА, то управление передается операторам, следующим за первым словом THEN, а затем к строке следующей за разделителем «;». Если - ЛОЖЬ, то определяется результат следующего ло-

*При записи структуры операторов в общем виде элементы, заключенные в «[]», являются необязательными.

гического выражения, и в случае ИСТИНЫ управление передается операторам, записанным за следующим THEN, а потом к строке идущей за «;» и т.д. Если же ни одно из условий оператора не выполняется, то выполняются операторы, записанные после слова ELSE, а затем управление передается строке, следующей за «;».

В командах разрешено использовать только один оператор, но если вам необходимо написать более одного оператора в команде, применяют конструкцию **begin ... end;**, которая позволяет записать любое количество операторов.

Например: **If a>=d Then begin b :=4; c := 5;
d := a+b-c; End;**

Логические выражения состоят из числовых или текстовых данных, знаков отношений и логических операций (табл. 2).

Таблица 2

Логические операции

ЗНАКИ СРАВНЕНИЯ		ЛОГИЧЕСКИЕ ОПЕРАЦИИ	
Название знака	В программе	Название операции	В программе
Равно	=	Логическое умножение	AND
не равно	<>	Логическое сложение	OR
больше	>	Отрицание	NOT
больше или равно	>=		
меньше	<		
меньше или равно	<=		

Пример 20: Даны **a** и **b**. Найти **max (a, b)**.

Program zadacha;

Var

a, b : real;

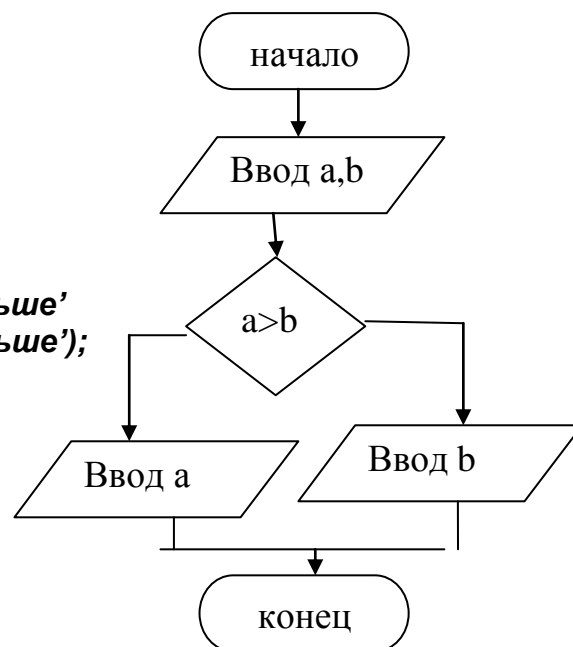
Begin

writeln('введите любые 2 числа');

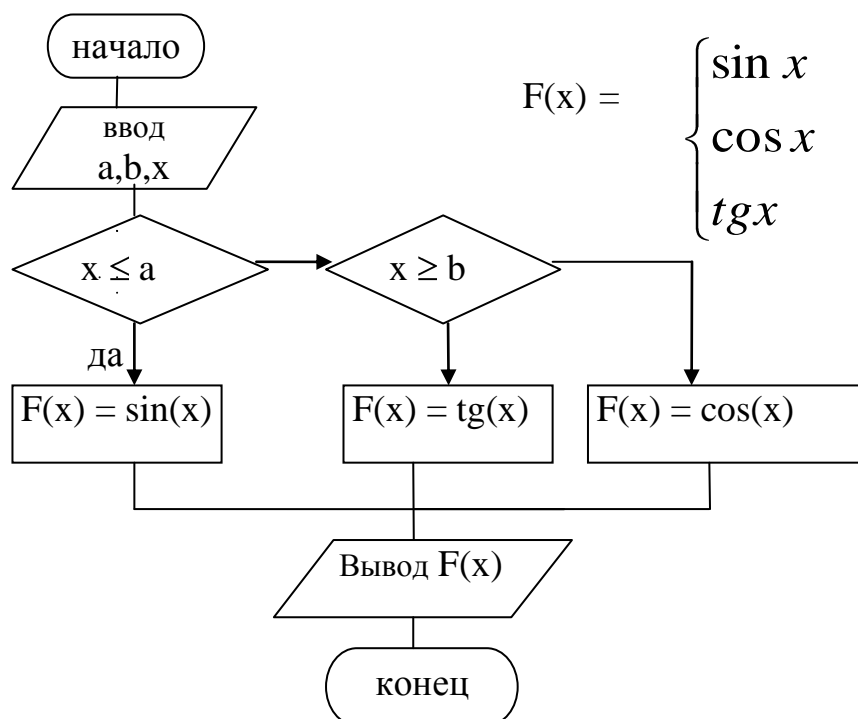
readln(a, b);

if a>b Then writeln('число', a, 'больше'
Else writeln ('число', b, 'больше');

end.



Пример 21: Выполнить вычисления



, если $x \leq a$
 , если $a < x < b$
 , если $x \geq b$

$$F(x) = \begin{cases} \sin x \\ \cos x \\ \operatorname{tg} x \end{cases}$$

Program zadacha;

Var

a, b, x, f: real;

Begin

writeln('введите любые 3 числа');

readln(a, x, b);

if (x > a) and (x < b) Then f := cos(x)

Else if x <= a Then f := sin(x)

Else f := cos(x)/sin(x);

writeln ('f=', f);

end.

Варианты задания 8

Составить программу вычисления и печати значений следующих выражений для исходных данных (данные подобрать самостоятельно), вводимых с клавиатуры ПК в диалоговом режиме:

$$F(x,y) = \begin{cases} 3x+1, & \text{если } x < 0 \\ y + \cos(x), & \text{если } x = 0 \\ y - \sin(x), & \text{если } x > 0 \end{cases}$$

$$F(x,y) = \begin{cases} y \cdot x, & \text{если } x \leq -1,75 \\ 74, & \text{если } -1,75 < x < 0,28 \\ 3x + x - 6, & \text{если } x \geq 0,28 \\ |x| + y, & \text{если } x < y \end{cases}$$

$F(x,y) =$	$74x \cdot (y-9x)$	в противном случае
$F(x,y) =$	$Y^2 - 5x,$	если $2x + 3 > 0$
	$74y,$	если $2x + 3 = 0$
	$3x + x - 6,$	если $2x + 3 < 0$
$F(x,y) =$	$\max(x, y+5),$	если $x > y$
	$(x+1) \cdot y + 3,$	если $x < y$
	$x \cdot y,$	если $x = y$
$F(x,y) =$	$x + \sin(y),$	если $x - y > 0$
	$y - \cos(x),$	если $x - y < 0$
	$\text{tg}(x) + 4y,$	если $x = y$
$F(x,y) =$	$1,$	если $x < 1$
	$x + 5,$	если $1 < x < 2$
	$6y \cdot 2x + 4,$	если $2 < x < 4$
$F(x,y) =$	$X^2 + y^2,$	если $x > y$
	$x \cdot y,$	если $x < y$
	$ 2x - y ,$	если $x = y$
$F(x,y) =$	$x(x^2 + 1),$	если $x > y$
	$x - \cos(xy),$	если $x < y$
	$7 + y,$	если $x = y$
$F(x,y) =$	$x \cdot y / (x+y),$	если $x > y$
	$6y \cdot 2x + 4,$	если $x < y$
	$x + \sin(y),$	если $x = y$
$F(x,y) =$	$\min(x, 2y, 23),$	если $x > y$
	$(2x, y),$	если $x < y$
	$x + 1/y,$	если $x = y$
$F(x,y) =$	$Y^2 - 5x,$	если $x > y$
	$(2x - 1) \cdot y,$	если $x < y$
	$578,$	если $x = y$
$F(x,y) =$	$x^2 + 1/y,$	если $x > y$
	$6y \cdot 2x + 4,$	если $x < y$
	$x + \sin(y),$	если $x = y$
$F(x,y) =$	$\min(x, 2.5y),$	если $x > y$
	$ x - y ,$	если $x = y$
	$x \cdot y / (x+y),$	если $x < y$

$$\begin{aligned}
 F(x,y) &= \begin{cases} Y^3 - 5x^3, & \text{если } x > y \\ 8y - 7x, & \text{если } x < y \\ \cos(x) + y, & \text{если } x = y \end{cases} \\
 F(x,y) &= \begin{cases} x^2 - 9, & \text{если } x > y \\ 6y \cdot 2x + 41, & \text{если } x < y \\ 2x + \sin(y), & \text{если } x = y \end{cases} \\
 F(x,y) &= \begin{cases} \ln(x+y), & \text{если } x > y \\ \max(x, 4y), & \text{если } x < y \\ 1/x + 1/y, & \text{если } x = y \end{cases} \\
 F(x,y) &= \begin{cases} 5 - y, & \text{если } x < y \\ \lg(x \cdot y), & \text{если } x = y \\ |\min(x, 2y)|, & \text{если } x > y \end{cases} \\
 F(x,y) &= \begin{cases} 1/x + 1/y, & \text{если } x - 0,5 > y \\ 1/x + 1/y, & \text{если } x - 0,5 > y \\ 6y \cdot 2x + 4, & \text{если } x < y + 0,5 \end{cases} \\
 F(x,y) &= \begin{cases} 8^{x/3} + y^2, & \text{если } x > y \\ \ln(x+1), & \text{если } x < y \\ y - 5x, & \text{если } x = y \end{cases} \\
 F(x,y) &= \begin{cases} 4x - 1y - 24, & \text{если } x > y \\ 6y \cdot 2x + 4, & \text{если } x < y \\ x + \sin(y), & \text{если } x = y \end{cases} \\
 F(x,y) &= \begin{cases} (x+1) \cdot (2y, 3), & \text{если } x > y \\ \max(x+5, y), & \text{если } x < y \\ (x/3)^2, & \text{если } x = y \end{cases} \\
 F(x,y) &= \begin{cases} y^2 - 1x, & \text{если } 2x > y \\ 3x \cdot \sin(y), & \text{если } 2x < y \\ 5y \cdot \cos(x), & \text{если } 2x = y \end{cases} \\
 F(x,y) &= \begin{cases} 3x \cdot |\sin(x)|^2, & \text{если } x > 6,5y \\ 6y \cdot 2x + 4, & \text{если } x < 6,5y \\ x + \sin(y), & \text{если } x = 6,5y \end{cases} \\
 F(x,y) &= \begin{cases} \operatorname{arctg}(x), & \text{если } x > y \\ x \cdot (4-y), & \text{если } x < y \\ xy + 4xy, & \text{если } x = y \\ x^2 + y^2 - 4xy, & \text{если } x > y \end{cases}
 \end{aligned}$$

$$F(x,y) = \begin{cases} x+1, & \text{если } x < y \\ y - 5x, & \text{если } x = y \end{cases}$$

$$F(x,y) = \begin{cases} |(x+y)| - 4,2, & \text{если } x > y \\ (x+y) \cdot 5, & \text{если } x < y \\ (x+5+y^2+3), & \text{если } x = y \end{cases}$$

$$F(x,y) = \begin{cases} (x \cdot y), & \text{если } x > y \\ (y \cdot x) \cdot y + 5, & \text{если } x < y \\ xy + (x+y), & \text{если } x = y \end{cases}$$

$$F(x,y) = \begin{cases} (2+x) \cdot y, & \text{если } x > 5 \\ \text{Cos}(x), & \text{если } -2 < x \leq 5 \\ Xy, & \text{если } x < -2 \end{cases}$$

$$F(x,y) = \begin{cases} 3x+y^2, & \text{если } x < 0 \\ 64, & \text{если } x = 0 \\ (x+y) \cdot x & \text{если } x > 0 \end{cases}$$

Задание 9. Программирование циклических алгоритмов

Теоретические положения.

Алгоритм называется циклическим, если все или отдельные его этапы в процессе решения задачи неоднократно повторяются.

Цикл обеспечивает повторное выполнение или, иначе говоря, циклическую работу операторов. Оператор или группа операторов, повторяющаяся в цикле, называется "телом цикла".

Далее рассмотрим два типа циклических задач:

а) задачи, в которых вычисления многократно ведутся по одним и тем же формулам с различными значениями входящих в нее величин;

б) задачи, где значение некоторой величины вычисляется через значение этой же величины, полученное в предыдущем цикле (рекурсии). Примерами таких задач являются задачи вычисления сумм и произведений рядов, а также вычисление значений факториала.

Характерные моменты циклического алгоритма:

- первоначальный вход в цикл выполняется через блок подготовки;
- цикл всегда характеризуется некоторой переменной, называемой параметром цикла. Начальное значение параметра задается перед циклом в блоке подготовки, а при каждом повторении цикла выполняются операторы тела цикла и начальное значение параметра изменяется на единицу;
- число повторений цикла должно быть конечным, однако, не всегда число повторений известно или может быть вычислено заранее. В этом случае

выход из цикла осуществляется при выполнении некоторых условий. Когда число повторений известно или может быть определено заранее, выход из цикла осуществляется при достижении параметром некоторой заранее заданной величины. Для такого рода задач используется оператор цикла "**FOR**". В общем виде оператор цикла записывается следующим образом:

For I := n TO k do <оператор1>;

или

For I := n TO k do begin

<оператор1>;

<оператор2>;

.....

<оператор N>; End;

Здесь:

I - параметр цикла (переменная),

n - начальное значение параметра цикла (переменная или число),

k- конечное значение параметра цикла (переменная или число).

Этот оператор многократно выполняет <оператор>, находящийся после ключевого слова **do** для всех значений параметра **I** от **n** до **k** (при этом $I = I + 1$, $n < k$).

Внимание. Если параметр **TO** заменить на параметр **DOWNTO**, то значение параметра **I** вычисляется по формуле $I = I - 1$ ($n > k$).

Структура самого оператора включает и подготовку цикла, и изменение параметра, и проверку условия выхода из цикла.

Проиллюстрируем использование оператора "**For**" в следующих заданиях.

Когда условие задачи не позволяет определить заранее число повторений цикла, выход из цикла зачастую осуществляется по достижении заданной точности вычислений или по какому-либо другому условию, оговоренному в задании. В этих случаях удобно использовать следующие операторы цикла:

While L do <оператор >;	Repeat < операторы >; Until L
--------------------------------------	--

Здесь: L - логическое выражение.

Операторы, находящиеся после **do** и после **Repeat** повторяются до тех пор, пока выражение, стоящее после **While**, - истинно или до тех пор, пока выражение, стоящее после **Until**, - ложно.

В качестве данных - исходных, промежуточных или результирующих, могут быть как простые переменные, так и элементы массивов.

Пример 22: Просуммировать $S = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$

n – задаем с клавиатуры

Program zadacha_12;

Var

```

s : real;           i, n: integer;
Begin
  writeln('введите любое число n');
  readln(n);
  s:=0;
  For i:=1 to n do s:=s+1/i;
  writeln('s=', s);
end.

```

Варианты задания 9*.

Написать программу вычисления и схему алгоритма

1. $S = 1 - \frac{1}{1 \cdot 2} + \frac{1}{1 \cdot 2 \cdot 3} - \frac{1}{1 \cdot 2 \cdot 3 \cdot 4} + \frac{1}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5} - \dots + (-1)^n \frac{1}{(N+1)!}$
2. $S = 1 - \frac{4}{7} + \frac{4}{11} - \frac{4}{15} + \frac{4}{19} - \frac{4}{23} + \dots$
3. $S = 1 + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{9} + \dots + \frac{1}{2N+1}$
4. $S = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \frac{1}{16} - \dots + (-1)^n \frac{1}{2^n}$
5. $S = \frac{5}{3} + \frac{5}{6} - \frac{5}{9} + \frac{5}{12} - \frac{5}{15} + \dots$
6. $S = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots + (-1)^n \frac{1}{2N+1}$
7. $S = 1 + \frac{1}{2} - \frac{1}{5} + \frac{1}{8} - \frac{1}{11} + \dots - (-1)^n \frac{1}{2+3N}$
8. $S = -\frac{7}{1} + \frac{7}{5} - \frac{7}{9} + \frac{7}{13} - \frac{7}{17} + \dots$
9. $S = -\frac{2}{4} + \frac{2}{7} - \frac{2}{10} + \frac{2}{13} - \frac{2}{16} + \dots$
10. $S = 1 + \frac{1}{2} + \frac{1}{5} + \frac{1}{8} + \frac{1}{11} + \dots + \frac{1}{2+3N}$
11. $S = -\frac{1}{1} + \frac{1}{5} + \frac{1}{9} + \frac{1}{13} + \frac{1}{17} + \dots$
12. $S = -\frac{3}{2 \cdot 4} + \frac{3}{2 \cdot 7} - \frac{3}{2 \cdot 10} + \frac{3}{2 \cdot 13} - \frac{3}{2 \cdot 16} + \dots$

* При программировании задач данного типа важно правильно задать начальные значения переменных. Для простых формул не приведена формула ряда в общем виде и ее следует вывести самостоятельно.

13. $S = 1 + \frac{1}{1 \cdot 2} + \frac{1}{1 \cdot 2 \cdot 3} + \frac{1}{1 \cdot 2 \cdot 3 \cdot 4} + \frac{1}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5} + \dots + \frac{1}{N!}$
14. $S = 4 + \frac{4}{4} + \frac{4}{9} + \frac{4}{16} + \frac{4}{25} + \frac{4}{36} + \dots + \frac{4}{N^2}$
15. $S = 1 + \frac{7}{2} + \frac{7}{4} + \frac{7}{8} + \frac{7}{16} + \dots + \frac{7}{2^n}$
16. $S = \frac{3}{2} + \frac{3}{5} - \frac{3}{8} + \frac{3}{11} - \frac{3}{14} + \dots$
17. $S = \frac{2}{2} + \frac{2}{5} + \frac{2}{8} + \frac{2}{11} + \dots + \frac{2}{2+3N}$
18. $S = \frac{13}{1} + \frac{13}{5} + \frac{13}{9} + \frac{13}{13} + \frac{13}{17} + \dots$
19. $S = \frac{1}{3} + \frac{1}{8} + \frac{1}{15} + \frac{1}{24} + \dots + \frac{1}{(N-1)(N+1)}$
20. $S = 1 + \frac{1}{1 \cdot 2} + \frac{2}{1 \cdot 2 \cdot 3} + \frac{3}{1 \cdot 2 \cdot 3 \cdot 4} + \frac{4}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5} + \dots + \frac{(N-1)}{N!}$
21. $S = 1 + 7 + 7^2 + 7^3 + 7^4 + \dots$
22. $S = 1 + \frac{2}{1} + \frac{3}{1 \cdot 2} + \frac{4}{1 \cdot 2 \cdot 3} + \dots + \frac{N}{(N-1)!}$
23. $S = 1 + \frac{2}{1 \cdot 2} + \frac{3}{1 \cdot 2 \cdot 3} + \frac{4}{1 \cdot 2 \cdot 3 \cdot 4} + \dots + \frac{N}{N!}$
24. $S = \frac{1}{3} - \frac{1}{9} + \frac{1}{27} - \frac{1}{81} + \dots - (-1)^{n-1} \frac{1}{3^n}$
25. $S = 1 + \frac{1}{2} - \frac{1}{7} + \frac{1}{12} - \frac{1}{17} + \dots - (-1)^n \frac{1}{2+5N}$
26. $S = \frac{5}{1} + \frac{5}{5} + \frac{5}{9} + \frac{5}{13} + \frac{5}{17} + \dots$
27. $S = -\frac{2}{4} + \frac{2}{9} - \frac{2}{14} + \frac{2}{19} - \frac{2}{24} + \dots$
28. $S = 3 + \frac{3}{1 \cdot 2} + \frac{3}{1 \cdot 2 \cdot 3} + \frac{3}{1 \cdot 2 \cdot 3 \cdot 4} + \frac{3}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5} + \dots + \frac{3}{N!}$
29. $S = 1 + \frac{3}{6} + \frac{3}{11} + \frac{3}{16} + \frac{3}{21} + \frac{3}{26} + \dots$
30. $S = 1 + \frac{1}{3} + \frac{1}{6} + \frac{1}{9} + \frac{1}{12} + \dots + \frac{1}{3N}$

ПРИЛОЖЕНИЕ А

ДВОИЧНО- ВОСМЕРИЧНАЯ ЗАВИСИ- МОСТЬ		
10-е число	8-е число	2-я триада
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111

ДВОИЧНО- ШЕСТНАДЦАТИРИЧНАЯ ЗАВИ- СИМОСТЬ		
10-е число	16-е число	2-я тетра- да
0	0	0000
1	1	0001
2	3	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Содержание

Введение.....	3
ЧАСТЬ I. Методические указания по изучению дисциплины.....	4
ЧАСТЬ II. Методические указания по выполнению контрольных работ.....	17
1. Выбор вариантов заданий контрольной работы.....	18
ЗАДАНИЕ 1. ПЕРЕВОД ЧИСЕЛ ИЗ ДЕСЯТИЧНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В СИСТЕМУ СЧИСЛЕНИЯ С ОСНОВАНИЕМ "в".	18
ВАРИАНТЫ ЗАДАНИЯ № 1.	20
ЗАДАНИЕ 2. ПЕРЕВОД ЧИСЕЛ ИЗ СИСТЕМЫ СЧИСЛЕНИЯ С ОСНОВАНИЕМ "в" В ДЕСЯТИЧНУЮ СИСТЕМУ СЧИСЛЕНИЯ	20
ВАРИАНТЫ ЗАДАНИЯ 2.	21
ЗАДАНИЕ 3. ПЕРЕВОД ЧИСЕЛ ИЗ СС С ОСНОВАНИЕМ 8 В СС С ОСНОВАНИЕМ 16 (ИЛИ ОБРАТНО).....	21
ВАРИАНТЫ ЗАДАНИЯ 3.	22
ЗАДАНИЕ 4. АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ В ДВОИЧНОЙ СС	23
ВАРИАНТЫ ЗАДАНИЯ 4	24
ЗАДАНИЕ 5. СЛОЖЕНИЕ ЦЕЛЫХ ДВОИЧНЫХ ЧИСЕЛ В ОБРАТНОМ ИЛИ ДОПОЛНИТЕЛЬНОМ КОДЕ	24
ВАРИАНТЫ ЗАДАНИЯ 5.	27
ЗАДАНИЕ 6. СЛОЖЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ	27
ВАРИАНТЫ ЗАДАНИЯ 6	29
ЗАДАНИЕ 7. ПРОГРАММИРОВАНИЕ ПРОСТЫХ ВЫЧИСЛЕНИЙ	29
ВАРИАНТЫ ЗАДАНИЯ 7	33
ЗАДАНИЕ 8. ПРОГРАММИРОВАНИЕ ВЕТВЯЩИХСЯ АЛГОРИТМОВ	34
ВАРИАНТЫ ЗАДАНИЯ 8	36
ЗАДАНИЕ 9. ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ	39
ВАРИАНТЫ ЗАДАНИЯ 9.	41
Приложение А	43