

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

Информатика

Методические указания к выполнению лабораторных работ

(Для заочной формы обучения)

1Часть

2014

Составители: М.В.Соколовская

Рецензент: к.т.н., доц. В.П.Попов

Приведены методические указания к выполнению контрольной и лабораторных работ по дисциплинам «Информатика», «Информационные технологии» для студентов заочной формы обучения. Дан краткий теоретический материал по тематике работ.

Могут быть полезны для студентов других форм обучения по другим дисциплинам. Приведенные материалы соответствуют ФГОС и рабочим программам указанных дисциплин.

Подготовлены к публикации кафедрой «Прикладная математика» и рекомендованы к изданию редакционно-издательским советом Санкт-Петербургского государственного университета аэрокосмического приборостроения.

ВЕДЕНИЕ

Целью проведения лабораторных работ является получение начальных практических навыков по получению, обработке, хранению и передаче информации при помощи современных технических средств и установленных на них приложений, необходимых студентам для правильного составления различных отчетов и освоения специальности.

В процессе изучения дисциплины студенты знакомятся с основными понятиями и терминологией информатики и информационных технологий, техническими и программными средствами их реализации.

Задачами лабораторных работ является изучение:

- основных понятий информатики;
- структуры и принципов работы ЭВМ как технического средства реализации информационных технологий;
- средств и алгоритмов представления, хранения и обработки информации;
- организации и средств человеко-машинного интерфейса.

Для успешного прохождения данных лабораторных работ по «Информатике» студент должен быть подготовлен по следующим дисциплинам:

- «Информатика» в пределах общеобразовательной программы средней школы;
- «Математика» в пределах общеобразовательной программы средней школы;
- «Дискретная математика», в разделах математической логики.

Для более углубленного усвоения материала курса и лабораторных работ, в конце указан список литературы.

Контрольная работа №1

ОПРЕДЕЛЕНИЕ КОЛИЧЕСТВА ИНФОРМАЦИИ, СОДЕРЖАЩЕГОСЯ В СООБЩЕНИИ

Цель работы. Получение практических навыков численного определения количества информации, содержащегося в сообщении.

Порядок выполнения лабораторной работы

Данная работа выполняется в файле электронной таблицы из любого прикладного офисного приложения Создать таблицу (50 рабочих строк), аналогичную таблице 1. Исходным материалом может являться любой технический текст объемом **не менее** 30 печатных страниц.

Таблица 1. Расчет энтропии источника

<i>№ n/n</i>	<i>Символ</i>	<i>Код символа</i>	<i>Число вхождений символа в текст</i>	<i>Вероятность вхождения символа (p_i)</i>	<i>I_i</i>
1	2	3	4	5	6
1	0				
2	1				
...	...				
50	я				
				(должна получиться «1»)	
↑ <i>Всего символов в тексте (K)</i>				↑ <i>Полная вероятность(P)</i>	↑ <i>Энтропия источника (I_{cp})</i>

Заполнить столбец **Символ** следующими значениями:

- 33 буквы русского алфавита;
- 10 цифр (0 — 9);
- Знаки препинания – «.», «,», «:», «;», «-», «пробел», «(»

(Примечание: Открывающая и закрывающиеся скобки имеют один и тот же код).

Заполнить столбец **Код символа** используя функцию «КОДСИМВ(...)»,

находящуюся в категории «Текстовые».

Открыв каскадом текст по варианту и таблицу и используя в Word «**Правка** ⇒ **Заменить**» заполнить столбец **Число вхождений символа в текст**. (Предполагается, что других символов в тексте НЕТ.) Сосчитать общее число символов.

По формулам заполнить столбцы « p_i » и « I_i ». Сосчитать полную вероятность и энтропию источника.

Создать таблицу, аналогичную таблице 2 и заполнить ее по формулам.

Таблица 2. Сравнение алфавитов

	Неопределенность	Разрядность кода	Абсолютная избыточность	Относительная избыточность
Стандартная кодовая таблица ASCII (кол-во символов)				
Созданная таблица (кол-во символов)				

Выписать применяемые формулы с расшифровкой используемых символов.

Содержание отчёта

1. Название и цель работы.
2. Заполненная таблица №1 для 50-ти символов.
3. Заполненная таблица №2.
4. Использованные формулы с определением переменных.
5. Выводы по работе соответственно цели лабораторной работы.
6. Сравнительный анализ таблицы на рис.2.

Приложение к лабораторной работе

Основные сведения об OpenOffice.org

OpenOffice.org — как следует из названия, открытый офисный пакет, предназначенный практически для любых задач художественного, вычислительного, алгоритмического направления (см.рис.1). Последние версии имеют название LibreOffice (в связи с особенностями лицензионного соглашения), не влияющее на свои основные функциональные возможности.



Рисунок 1. Приглашение в OpenOffice.org и LibreOffice.

Так как OpenOffice.org (в дальнейшем, для краткости, ООо) является открытой системой, то она хорошо встаёт на любую операционную систему. Лёгкость установки делает её одной из самых удобных интегрированных сред. Она сочетает в себе возможности систем управления базами данных, табличного и текстового процессоров, деловой графики и пр. Все компоненты интегрированной среды имеют схожий интерфейс, что очень облегчает профессиональному обучению с ними.

Приложения ООо содержат большое количество готовых шаблонов, а также мастеров для создания оригинальных документов и шаблонов. Но, если необходимо. Можно создавать свои собственные шаблоны с помощью мастера.

ООо содержит следующие основные модули (рис.1):

- **Writer** — редактор текстовых документов;

- *Calc* — редактор формул;
- *Draw* — редактор рисунков;
- *Impress* — создание презентаций;
- *Math* — создание формул;
- *Base* — работа с базами данных.

Каждое приложение можно интегрировать в другое приложение. Например, можно вставлять и редактировать рисунки, практически любого формата, в *Writer*; или составить презентацию — *Impress* из диаграмм и таблиц *Calc*.

Создана единая справочная система по всему ООо, которая вызывается из любого приложения, и, как принято в современных программах, имеет активные гиперссылки на необходимые подпункты.

Глоссарий по OpenOffice.org

Стандартные панели OpenOffice.org

Окно ООо содержит стандартные области (см.рис.2):

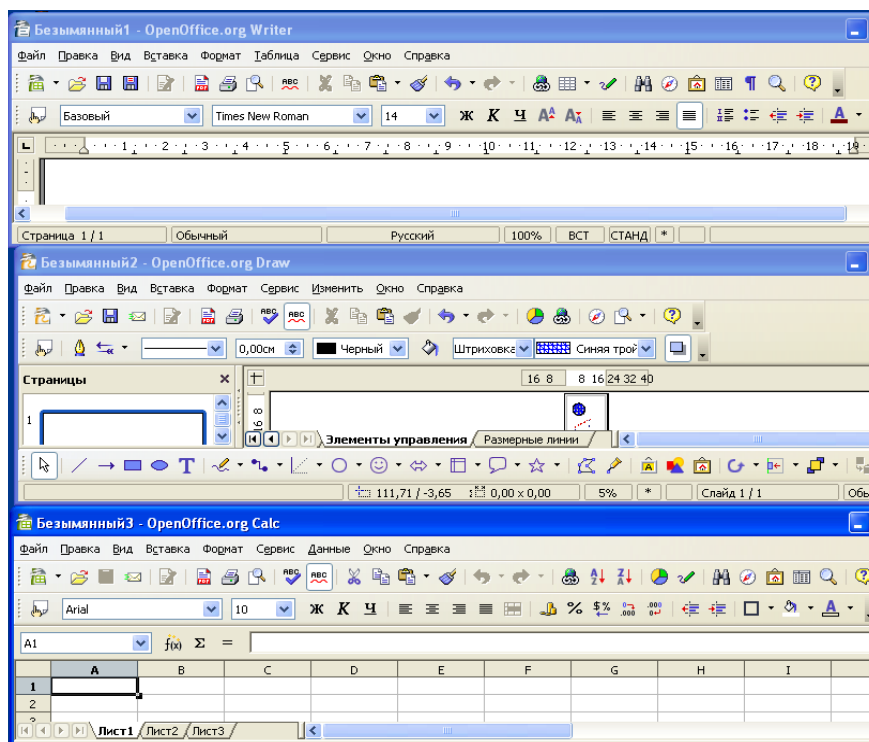


Рисунок 2. Общий вид окон OpenOffice.org.

- строка наименования;

- строка меню;
- стандартная панель инструментов;
- панель инструментов приложения;
- строка состояния;
- линии прокрутки — вертикальная и горизонтальная;
- рабочая область приложения.

Интерфейс всех областей приложения можно настраивать под индивидуального пользователя. Все области можно дополнять специальными знаками, предусмотренными возможностями ООо.

Работа в Microsoft Office и OpenOffice.org

В OpenOffice.org можно открывать и сохранять документы в форматах файлов Microsoft Office.

Открытие файла Microsoft Office. Выберите команду **Файл ⇒ Открыть**. В ООо, в диалоговом окне открытия файлов, выберите файл Microsoft Office.

Сохранение как файл Microsoft Office. Выберите команду **Файл ⇒ Сохранить как**. В открывшемся окне, в списке поля **Тип файла** выберите формат файла Microsoft Office.

Microsoft Office и ООо не могут выполнять одни и те же коды макросов. В Microsoft Office используется вариант языка VBA (Visual Basic for Applications), а в ООо используется Basic, основанный на среде OpenOffice.org API (программного интерфейса приложения). Хотя язык программирования один, но объекты и методы разные.

ООо может автоматически открывать документы Microsoft Office 97/2000/XP/2003, а так же новые версии Office Open XML (.docx). Однако некоторые особенности разметки и атрибуты форматирования сложных документов Microsoft Office в ООо обрабатываются по-другому или не поддерживаются. По этой причине преобразованные файлы необходимо в некоторой степени переформатировать вручную. Ожидаемые объемы

переформатирования пропорциональны сложности структуры и форматирования исходного документа.

Особенности преобразования Microsoft Office. Ниже представлен общий список особенностей Microsoft Office, которые могут вызвать проблемы преобразования. Они не влияют на возможность использования или работы с содержимым преобразованного документа:

- Автофигуры
- Пометки рецензирования
- Объекты OLE
- Некоторые элементы управления и поля форм Microsoft Office
- Указатели
- Форматирование таблиц, рамок и колонок
- Гиперссылки и закладки
- Графические объекты Microsoft WordArt
- Анимация символов/текста

При разработке документа Writer можно использовать полезные возможности, такие как: стиль и форматирование, проверка орфографии, автозамена, активные гиперссылки и закладки, шаблоны. Выведенный навигатор позволяет быстро перемещаться по редактируемому документу. Что значительно сокращает время поиска изменяемого блока документа.

Основные сведения об информации

Множество всех знаков и сигналов, использующееся для формирования сообщения, называется алфавит.

Размер (глубина) алфавита A определяется количеством символов, составляющих алфавит. Если считать, что сообщение передается одним знаком алфавита размером A , всего может быть передано $N=A$ сообщений.

Из знаков алфавита может быть составлено слово. Если размер слова фиксировано и составляет n знаков, то количество возможных слов N составленных символов из алфавита A , таким образом, что каждый символ алфавита может входить в слово $0,1,2,\dots,n$, раз определяется

$$N = A^n \quad (1)$$

Таким образом, с помощью слов можно представить информацию о любом из N сообщений.

Выражение (1) позволяет определить размер слова из алфавита A , с помощью которого можно представить N сообщений

$$n = \lceil \log_A N \rceil \quad (2).$$

Мы можем сопоставить тому или иному сообщению комбинацию знаков, тогда при приеме сообщения, зная правила сопоставления, можно распознать сообщение.

Важный вопрос теории передачи и преобразования информации — установление меры, количества и качества информации.

Математические меры информации. Информационные меры, как правило, рассматриваются в двух аспектах *синтаксическом* и *семантическом*.

Семантический подход позволяет выделить полезность или ценность информационного сообщения (в настоящем пособии не рассматривается).

В *синтаксическом* аспекте сообщения рассматриваются как символы, абстрагированные от содержания и какой-либо ценности. Предметом анализа и оценивания являются частота появления символов, связи между ними, порядок следования, правила построения сообщений. В таком рассмотрении наиболее широко используют *структурные* и *вероятностные* (статистические) меры.

Структурные меры оценивают строение массивов информации и их измерение простым подсчетом информационных элементов или комбинаторным методом. Структурный подход применяется для оценки возможностей информационных систем вне зависимости от условий их применения.

При статистическом подходе используется понятие *энтропии* как меры неопределенности, учитывающей вероятность появления и информативность того или иного сообщения. Статистический подход учитывает конкретные условия применения информационных систем.

При синтаксическом анализе информация определяется как *мера*

уменьшения неопределенности знаний о каком-либо предмете в познавательном процессе. Если H_1 — исходная (априорная) неопределенность до передачи сообщения, а H_2 — остаточная (апостериорная) неопределенность, характеризующая состояние знания после получения сообщения, то содержащаяся в этом сообщении информация определяется их разностью

$$I = H_1 - H_2. \quad (3)$$

Известно достаточно большое количество различных мер, различающихся подходом к определению неопределенности в (3). Далее рассматриваются только две из них — структурная аддитивная мера Хартли и вероятностная мера, называемая энтропией, предложенная К.Шенноном.

Структурная мера информации. Это понятие может быть распространено не только на число, но и на вариант не числового сообщения.

Аддитивная мера Хартли использует понятия глубины A и длины n сообщения. В этом случае глубина числа тождественна размеру алфавита, а длина числа — разрядности слова при передаче символьного сообщения.

Если сообщение — число; то понятие глубины числа будет трансформировано в понятие основания системы счисления. При заданных глубине и длине числа количество чисел, которое можно представить, $N = A^n$. Очевидно, что N однозначно характеризует степень исходной неопределенности. Исходная неопределенность по Хартли определяется

$$H_1 = \log_a N. \quad (4)$$

Неопределенность после получения сообщения, остаточная неопределенность,

$$H_2 = \log_a N^*, \quad (5)$$

где N^* — число возможных значений принятого слова после получения сообщения.

Основание логарифма в (5) определяет только единицы измерения неопределенности. При $a=2$ это двоичная единица информации, называемая бит. При $a = 10$ десятичная (*дит*), при $a = e$ натуральная (*нат*). Далее мы будем всегда пользоваться двоичной единицей информации.

$N^*=1$, если после получения информации нет неопределенности, т.е. получатель гарантировано получил то сообщение, которое было передано. Если в канале связи за счет влияния помех возникают искажения переданного сигнала, тогда получателю приходится, после приема информации, выбирать сообщения из некоторого множества, которое характеризует число возможных сообщений при выборе. Таким образом, если передается символ некоторого алфавита, то N^* определяет возможную неоднозначность приема этого символа за счет искажений в канале связи. В случае измерительного опыта, число N^* — характеризует число возможных значений величины после измерения и определяет погрешность измерения.

Очевидно, что должно быть $N^* < N$, а $N^* = 1$ только в идеальном случае передачи сообщения без потери информации или, что то же самое, измерения некоторой физической величины без ошибок. Количество информации по Хартли оценивается как

$$I=H_1 - H_2 = \log_a N - \log_a N^* = \log_a N/N^* . \quad (6)$$

Логарифмическая мера, позволяющая, вычислять количество информации, содержащейся в сообщении, переданном числом длиной n и глубиной A :

$$I = \log_2 N = n \log_2 A, \text{ бит}. \quad (7)$$

Следовательно, **1 бит** информации соответствует одному элементарному событию, которое может произойти или не произойти. Такая мера количества информации удобна тем, что она обеспечивает возможность оперировать мерой как числом. Из сравнения (7) и (2) следует, что численное значение неопределенности определяет число двоичных разрядов, необходимое для кодирования символа алфавита A .

Логарифмическая мера для неопределенности и информации выбрана не случайно. Она оказывается удобной при описании сложных опытов.

При наличии нескольких источников информации общее количество информации

$$I(q_1, q_2, \dots, q_n) = I(q_1) + I(q_2) + \dots + I(q_k), \quad (8)$$

где $I(q_k)$ — количество информации от источника q_k .

Логарифмическая мера информации позволяет измерять количество

информации и широко используется на практике. Всегда надо учитывать, что все сообщения в этой мере полагаются равновероятными и независимыми. Эти допущения приводит на практике к существенно завышенным оценкам.

Примечание. Для рассмотрения дальнейшего материала необходимо использовать понятие «**вероятность события**». Под вероятностью события принимается постоянная величина, около которой группируются значения частоты появления некоторого события, например, передачи одного из символов алфавита. Если частота появления любого символа алфавита при передаче длинной последовательности символов одинакова, то говорят о равновероятных событиях, символах, сообщениях и т.п. Независимыми сообщения полагают, если вероятности их передачи не зависят от того, какие сообщения были переданы ранее.

Статистическая мера информации. В статистической теории информации вводится более общая мера количества информации, в соответствии с которой рассматривается не само событие, а информация о нем. Этот вопрос глубоко проработан К. Шенноном в работе «Избранные труды по теории информации». Если появляется сообщение о часто встречающемся событии, вероятность появления которого близка к единице, то такое сообщение для получателя малоинформативное. Столь же мало информативны сообщения о событиях, вероятность появления которых близка к нулю.

События можно рассматривать как возможные исходы некоторого опыта, причем все исходы этого опыта составляют ансамбль, или полную группу событий. К. Шеннон ввел понятие неопределенности ситуации, возникающей в процессе опыта, назвав ее энтропией. Энтропия ансамбля есть количественная мера его неопределенности и, следовательно, информативности, количественно выражаемая как средняя функция множества вероятностей каждого из возможных исходов опыта.

Поясним содержание статистической меры на следующем частном случае. Пусть выполняется посимвольная передаче текста, состоящего из символов алфавита A . Текст составлен из K символов алфавита. Опыт состоит в передаче очередного символа текста. Так как в один момент времени может быть передан

любой символ алфавита, всего возможно A исходов опыта. Очевидно, что одни символы в тексте будут появляться часто, а другие — реже. Различные символы несут разную информацию. Обозначим через k_i количество появления символа в тексте, а количество вносимой этим символом информации как I_i . Будем полагать, что передаваемые символы независимы, т.е. передача i -того символа происходит с вероятностью, независящей от того, какой символ был передан ранее. Это означает, информация, вносимая символом постоянна для любых сочетаний символов. Тогда средняя информация, доставляемая одним опытом,

$$I_{cp} = (k_1 I_1 + k_2 I_2 + \dots + k_A I_A) / K. \quad (9)$$

Но количество информации в каждом исходе связано с его вероятностью p_i , и выражается в двоичных единицах (битах) как

$$I_i = \log_2 (1/p_i) = -\log_2 p_i .$$

Тогда

$$I_{cp} = [k_1 (-\log_2 p_1) + \dots + k_A (-\log_2 p_A)] / K. \quad (10)$$

Выражение (10) можно записать также в виде

$$I_{cp} = k_1 / K (-\log_2 p_1) + \dots + k_A / K (-\log_2 p_A). \quad (11)$$

Но отношения n/K представляют собой частоты повторения исходов, а, следовательно, могут быть заменены их вероятностями:

$$p_i = k_i / K ,$$

Тогда средняя информация в битах

$$I_{cp} = p_1 (-\log_2 p_1) + \dots + p_A (-\log_2 p_A),$$

или

$$I_{cp} = \sum p_i (-\log_2 p_i) = H \quad (12)$$

Полученную величину H называют *энтропией*. Энтропия обладает следующими свойствами:

Энтропия ВСЕГДА НЕОТРИЦАТЕЛЬНА, так как значения вероятностей выражаются величинами, не превосходящими единицу, а их логарифмы — отрицательными числами или нулем, так что члены суммы (12) — неотрицательны.

Случай, когда об опыте или величине все известно заранее и результат не дает

новую информацию что соответствует полной невозможности или полной достоверности события, энтропия равна нулю..

Энтропия имеет наибольшее значение, когда все вероятности равны между собой:

$$p_1 = p_2 = \dots = p_i = 1/A.$$

При этом $H = -\log_2(1/A) = \log_2 A = H_{max}.$

Если все события равновероятны и статистически независимы, то оценки количества информации, по Хартли и Шеннону, совпадают. Это свидетельствует о полном использовании информационной емкости системы. В случае неравных вероятностей количество информации, по Шеннону, меньше информационной емкости системы.

Абсолютная избыточность информации $D_{авс}$ представляет собой разность между максимально возможным количеством информации и энтропией:

$$D_{авс} = H_{max} - H. \quad (13)$$

Пользуются также понятием *относительной избыточности*

$$D = (H_{max} - H) / H_{max}. \quad (14)$$

Рассмотренные информационные меры в полной мере применимы для оценки количества информации при передаче и хранении информации в вычислительных системах и цифровых системах связи.

Информационные меры имеют важное значение при определении характеристик памяти ЭВМ, пропускной способности каналов связи и во многих других приложениях информатики.

Лабораторная работа №1

РАБОТА В КОМАНДНОЙ СТРОКЕ

Цель работы: изучить базовые команды DOS и Linux, получить основные навыки работы в командной строке операционной системы Windows и Ubuntu, а так же Norton Commander.

Задание и порядок выполнения лабораторной работы

Лабораторная работа выполняется в два этапа, соответственно двум изучаемым системам: Windows и Ubuntu по плану, указанному ниже.

1. Открыть командное окно, перейти в корневой каталог (внешнее запоминающее устройство).
2. Создать папку с названием системы и перейти в нее.
3. Поменять цвет экрана и шрифта.
4. В корневом каталоге создать три каталога: КАТ1, КАТ2, КАТ3 – **НАЗВАНИЯ КАТАЛОГОВ, ПОДКАТАЛОГОВ И ФАЙЛОВ ПРОИЗВОЛЬНЫ.**
5. Перейти в каталог КАТ2 и создать в нем подкаталоги — РКАТ1, РКАТ2.
6. Перейти в подкаталог РКАТ2; создать в нем четыре текстовых файла:
СТ1.txt — содержащий Имя,
СТ2.txt — содержащий соответственно Фамилию,
СТ3.txt — Отчество,
СТ4.txt — № группы. (Для более презентабельного вида в конце\начале каждого файла вставьте пробел.)
7. Вернуться в корневой каталог и создать в нем два текстовых файла: Спец.txt и Номер.txt, содержащих соответственно название специальности и № студенческого билета.
8. Поменять атрибуты файлов, сделав их только для чтения.
9. Создать в КАТ1 файл «СТУДЕНТ», являющийся объединением файлов СТ1 — СТ4 и одновременно удалив использованные файлы.
10. Записать в файл ...:\КАТ1\РКАТ\COLOR.txt при помощи переадресации, справку по изменению цветов.

11. Записать в файл ...:\KAT1\PKAT\DIRECT.txt при помощи переадресации, справку, содержащую дерево каталогов корневого каталога.

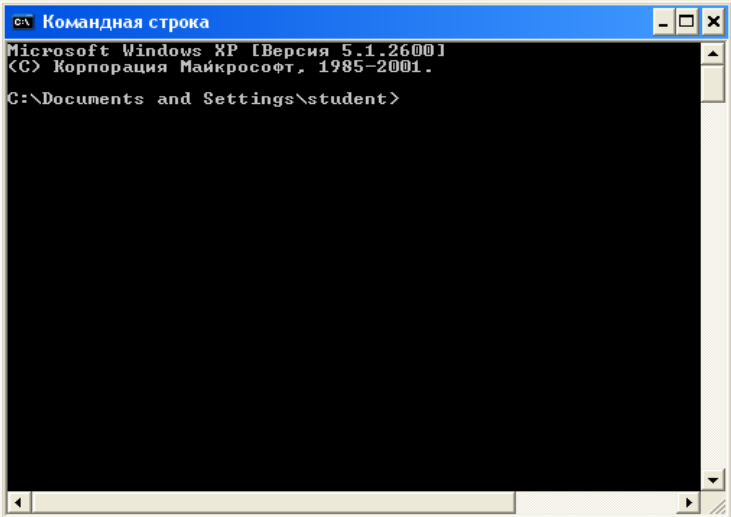

12. Предъявить выполненную работу преподавателю.

Письменный отчет НЕ оформляется.

Приложение к лабораторной работе

Для более наглядного представления сведений материал представлен в виде сравнительной таблицы 3.

Таблица 3. Базовые команды

Базовые команды DOS	Базовые команды Linux
Внешний вид рабочего окна	
	
Основные сведения	
<p>Приглашение. Когда DOS готова к работе с пользователем, она выдает на экран приглашение:</p> <ol style="list-style-type: none"> 1. C:\> — дисковод C: корневой каталог; 2. G:\КАТ> — дисковод G:, каталог КАТ. 	<p>Запуск эмулятора терминала в Ubuntu. Для этого нужно выбрать в меню Приложения⇒Стандартные⇒Терминал. Тем самым, вы запустите «командную строку», в которой вы можете вводить команды. Когда командный интерпретатор ожидает ввода, он выводит на экран приглашение, имеющее по умолчанию вид:</p>

<i>Базовые команды DOS</i>	<i>Базовые команды Linux</i>
	<i><пользователь>@<компьютер> <текущий каталог>\$</i>
<p>Ввод команд. Для ввода команды необходимо её набрать на клавиатуре и нажать <Enter>. Для прекращения выполнения любой команды необходимо нажать комбинацию клавиш <Ctrl> + <C> или <Ctrl> + <Break (пробел)>.</p> <p>Вызов в командную строку предыдущей команды — F3. Можно воспользоваться стрелкой вверх на клавиатуре. Удаление текущего или предыдущего символа — , <Backspace>. Очистка всей командной строки — <Esc>.</p>	<p>Ввод команд. Для ввода команды необходимо её набрать на клавиатуре и нажать <Enter>. Для прекращения выполнения любой команды необходимо нажать комбинацию клавиш <Ctrl> + <C> или <Ctrl> + <Z>.</p> <p>Вызов в командную строку предыдущей команды — <стрелка вверх>.</p> <p>Удаление текущего или предыдущего символа — , <Backspace>. Очистка всей командной строки — <Ctrl> + <U>.</p>
<p>Если выполняемая команда выдает слишком много информации на экран, можно воспользоваться комбинацией клавиш <Ctrl> + <S> для приостановки выдачи. Повторное нажатие комбинации возобновит выдачу информации. Для других программ выдачу</p>	<p>Если выполняемая команда выдает слишком много информации на экран, можно воспользоваться комбинацией клавиш <Ctrl> + <S> для приостановки выдачи. Повторное нажатие любой клавиши возобновит выдачу информации.</p>

<i>Базовые команды DOS</i>	<i>Базовые команды Linux</i>
<p>информации на экран можно приостановить нажатием комбинации клавиш <Ctrl> + <NumLock>.</p>	<p>Автозавершение позволяет при нажатии клавиши <Tab> автоматически дописывать начатую команду. Это очень удобное свойство командной оболочки экономит время при вводе. Бывает, мы не знаем, что вводить далее после нескольких букв, и командная оболочка после одиночного нажатия клавиши табуляции ничего не дополняет. В таком случае может помочь двойное нажатие Tab, после которого на экране появляется перечень всех возможных вариантов.</p>
<p>Запуск и выполнение команд. Для выполнения любой команды MS DOS необходимо ввести имя этой команды и её параметры.</p> <p>Для выполнения программ или командных файлов (файлы с расширением .COM, .EXE, .BAT) необходимо ввести имя соответствующего файла и параметры.</p> <p>Если программа не находится в текущем каталоге, то необходимо в качестве имени команды использовать полное имя файла, включающее путь к тому каталогу, в котором находится нужная программа или командный</p>	<p>Запуск и выполнение команд. Для выполнения любой команды необходимо ввести имя этой команды и её параметры. При вводе команд в UNIX необходимо быть внимательным к пробелам и регистру. Многие команды имеют дополнительные опции. После ввода команды необходим пробел перед вводом опции или имени файла. После ввода опции необходим также пробел. Команды и опции чувствительны к регистру.</p> <p>Для выполнения программ или командных файлов необходимо ввести имя соответствующего файла и</p>

<i>Базовые команды DOS</i>	<i>Базовые команды Linux</i>
<p>файл.</p> <p>Во многих командах MS DOS в именах файлов можно употреблять символы * и ? для указания группы файлов из одного каталога. Символ * обозначает ЛЮБОЕ количество символов в имени файла или в расширении имени файла. Символ ? обозначает ОДИН произвольный символ или отсутствие символа в имени файла или в расширении имени файла.</p> <p>Примеры:</p> <ul style="list-style-type: none"> ✓ *.bac — все файлы с расширением .bac из текущего каталога; ✓ c*.d* — все файлы с именем, начинающимся с «С», и расширением, начинающимся с «D», из текущего каталога; ✓ a:\doc\ABC???.* — все файлы с именем, начинающимся с ABC и состоящим не более чем из 6 символов, из каталога \doc на диске на дисковом A:. 	<p>параметры. Если программа не находится в текущем каталоге, то необходимо в качестве имени команды использовать полное имя файла, включающее путь к тому каталогу, в котором находится нужная программа или командный файл.</p> <p>Если в командной строке присутствует символ *, то командный интерпретатор заменяет фрагмент, содержащий * на список всех файлов, имеющихся в файловой системе, имена которых соответствуют данному фрагменту, считая, что * может соответствовать любое количество символов в имени файла.</p>
<p>Получение справки о командах. Для получения краткой справки по программе или команде MS DOS возможно, если ввести эту команду (или программу) с</p>	<p>Получение справки о командах. Краткую справку о применении любой команды можно получить если ввести в командной строке выражение: <имя</p>

<i>Базовые команды DOS</i>	<i>Базовые команды Linux</i>
<p>параметром \?. На экран будет выведена справка о назначении программы и формате её вызова. В MS DOS версий 5 и 6 на экран можно выдать список команд с кратким описанием их назначения. Для этого надо ввести команду: FASTHELP или FASTHELP <i>имя команды</i>.</p> <p>В MS DOS версии 6 имеется интерактивный справочник по командам и драйверам DOS. Для вызова справочника надо ввести команду: HELP или HELP <i>имя команды</i>.</p>	<p><i>команды</i>> — help.</p> <p>Страницы системной документации также выводит системная утилита man. С ее помощью можно получить информацию по любым командам Linux. Для этого используется синтаксис man <<i>имя команды</i>>. Например, чтобы вывести руководство по команде ls или cat используйте команды man ls и man cat соответственно.</p>
<p>«Зависание» компьютера. иногда выполняемая под управлением MS DOS программа начинает работать неправильно или же не реагирует на нажатия клавиш. В этом случае выполнение программы нужно прекратить нажатием сочетанием клавиш <Ctrl> + <Break (пробел)>. Если это не помогает, то надо перезагрузить DOS. Для этого следует одновременно нажать сочетание <Ctrl> + <Alt> + .</p>	
<p>Изменение цвета. Команда для изменения цвета фона окна и цвета шрифта color <<i>№ цвета фона</i>><<i>№</i></p>	<p>Изменение цвета. Команда tput setb <<i>№ цвета</i>> по 8-ричной системе задает цвет фона консоли.</p>

Базовые команды DOS	Базовые команды Linux
цвета шрифта>. Шкала цветов по 16-ричной системе.	Команда tput setf <№ цвета> задает цвет шрифта.
Работа с файлами	
<p>Создание текстовых файлов. Небольшие текстовые файлы можно набрать непосредственно с клавиатуры. Для этого необходимо ввести команду: сору <i>имя файла</i>. После ввода этой команды нужно будет поочередно вводить строки файла.</p> <p>В конце каждой строки надо нажимать <Enter>, а после ввода всего текста — нажать F6 и затем <Enter> (можно использовать сочетание: <Ctrl> + <Z>).</p>	<p>Создание текстовых файлов. Небольшие текстовые файлы можно набрать непосредственно с клавиатуры. Для этого необходимо ввести команду: > <i>имя файла</i>. Например, создадим текстовый файл <i>a.txt</i> :</p> <pre> > a.txt cat > a.txt - открытие пустого файла a.txt [Курсор переводится на следующую строку. Ожидается ввод с клавиатуры] first string <Enter> - ввод 1-й строки second string <Enter> - ввод 2-й строки < ctrl + D > - закрытие файла a.txt Добавление информации в файл <i>a.txt</i> производится с помощью следующего выражения: cat >> a.txt </pre>
<p>Удаление файлов. Для удаления файлов имеется команда del <i>имя файла</i>. В имени файла можно употреблять символы * и ? если необходимо удалить несколько файлов.</p>	<p>Удаление файлов. Для удаления файлов имеется команда rm <<i>имя файла</i>> (от англ. «remove» — «удалять»). В имени файла можно употреблять * и ?, которые заменяют собой любой символ.</p>
<p>Переименование файлов. Для переименования файлов применяется (RENAME) ren <i>старое имя файла</i></p>	

<i>Базовые команды DOS</i>	<i>Базовые команды Linux</i>
<p>новое имя файла2. В этой команде также можно пользоваться символом *. Старое и новое имена файлов должны быть разделены пробелом.</p>	
<p>Перемещение файлов в другой каталог. Для переноса файла с одного места в другое применяется команда move (/Y) имя файла имя каталога. В имени файла можно использовать * чтобы переместить несколько файлов в другой каталог. При перемещении файла в другой каталог его можно переименовать: move (/Y) имя файла (дисквод:)(путь) новое имя файла. При наличии (/Y) перезапись файлов выполняется без каких-либо запросов.</p>	<p>Перемещение и переименование файлов. Для перемещения, переименования файла предназначена утилита mv. Для переименования файла используется синтаксис mv <имя файла> <новое имя файла>. Для перемещения файла в другую директорию используется синтаксис mv <имя файла> <директория>.</p>
<p>Копирование файлов. При необходимости скопировать файл используется команда copy имя файла1 имя файла2. В этой команде можно использовать вместо имен файлов обозначения устройств, например PRN – принтер. Если копирование происходит в другой каталог, то необходимо прописать полный путь расположения файла. Имена файлов должны быть разделены пробелом.</p>	<p>Копирование файлов. Утилита cp используется для создания копии файла. Для обычного копирования используется следующий синтаксис cp <имя файла источника> <имя файла назначения>.</p>
<p>Соединение (конкатенация) файлов. Команда copy может использоваться также для</p>	<p>Соединение (конкатенация) файлов. Утилита cat также может использоваться для соединения,</p>

<i>Базовые команды DOS</i>	<i>Базовые команды Linux</i>
<p>объединения содержимого нескольких файлов в один файл: copy имя файла1 (+ имя файла2) ... имя файла. Имя нового файла должно быть отделено от исходных пробелом. Лучше предварительно создать пустой файл, и затем уже объединять файлы в пустой файл. Если копирование происходит в другой каталог, то необходимо прописать полный путь расположения файла.</p>	<p>конкатенации двух текстовых файлов, от англ. «concatenation» и происходит название утилиты cat. Например, у нас есть два файла <i>a.txt</i> и <i>b.txt</i>. Команда cat a.txt b.txt выведет на экран оба файла, а команда cat a.txt >>b.txt объединит оба файла в единый файл <i>b.txt</i>.</p>
<p>Поиск файлов на диске. Для поиска файлов на диске по имени файла и\или по содержащейся в файле строке символов можно использовать программу FileFind из комплекса Norton Utilities: filefind имя файла (строка символов) (режимы) — /C — режим поиска только в текущем каталоге, а если /C и /S — поиск в текущем каталоге и во всех его подкаталогах.</p>	<p>Поиск файлов на диске. Для поиска файлов из командной строки вы можете использовать команду find. У этой команды следующий синтаксис: find имя файла.</p> <p>Если не указан путь к директории файла, поиск идет в текущей директории.</p> <p>Команда locate ищет файлы по всей файловой системе. Синтаксис команды locate имя файла.</p> <p>Команда apropos ищет строки в заголовках и названиях документации (дополнительно вводится слово для поиска). Выдаёт список всего найденного.</p>
<p>Чтение файлов. Для открытия файлов в MS DOS для чтения используется команда type имя файла.</p>	<p>Вывод содержимого файлов. Выводить содержимое файла на экран можно несколькими</p>

<i>Базовые команды DOS</i>	<i>Базовые команды Linux</i>
	<p>способами.</p> <p>Команда less выводит содержимое файла на экран постранично. Смена страницы производится с помощью клавиши пробела. Выход из режима просмотра производится с помощью клавиши Q.</p> <p>Утилита cat также выводит содержимое текстового файла. Простейший синтаксис команды cat <i><имя файла></i>. Для вывода больших файлов используется команда cat <i><имя файла></i> more, которая будет выводить файл постранично.</p>
Работа с каталогами	
<p>Команда смены текущего каталога: A: — переход на дисковод A.</p>	
<p>Изменение текущего каталога: cd (<i>дисковод:</i>) <i>путь</i>. При необходимости подняться на один уровень вверх можно не указывать путь: cd.. .</p>	<p>Изменение текущего каталога. Команда cd предназначена для смены каталога, в котором вы сейчас находитесь. Например, если вы находитесь в своем домашнем каталоге и хотите перейти в подкаталог dir, тогда нужно просто выполнить команду cd dir. Домашний каталог в Linux обозначается знаком ~. Если выполнить</p>

<i>Базовые команды DOS</i>	<i>Базовые команды Linux</i>
	команду cd без параметров, то вы перейдете в свой домашний каталог .
<p>Просмотр каталога: <code>dir</code> (дисконд:) (путь\) (имя файла) (параметры). /P — по экранный вывод оглавления; /W — вывод данных в широком формате: приводятся только имена файлов и подкаталогов.</p>	<p>Просмотр каталога. Команда pwd отображает полный путь к текущему каталогу.</p>
<p>Вывод дерева каталога. Команда tree выводит дерево каталогов, находящихся ниже активного каталога.</p>	<p>Вывод содержимого каталога. Утилита ls предназначена, для вывода списка файлов и подкаталогов в текущем каталоге. Рассмотрим некоторые полезные опции для команды ls.</p> <ul style="list-style-type: none"> -a — включить в список «скрытые» файлы («скрытые» файлы в Linux имеют в своем названии точку в начале имени файла, например, .local). -s—отсортировать файлы по размеру. -t—отсортировать по дате изменения. -l—выводить имена в столбик, по одному на каждой строке. -l—вывести подробную информацию о каждом файле, размер файла, владелец файла, права доступа, дата

<i>Базовые команды DOS</i>	<i>Базовые команды Linux</i>
	<p>изменения.</p> <p>-R- вывести содержимое подкаталогов.</p>
<p>Создание каталога: Для создания нового каталога применяется команда md (дискковод:) <i>путь</i>.</p>	<p>Создание каталога. Команда mkdir предназначена для создания новой директории. Чтобы создать новую директорию, выполните mkdir <i><имя директории></i>. Для того, чтобы создать сразу несколько директорий, укажите их имена через пробел, например: mkdir dir1 dir2 dir3.</p>
<p>Уничтожение каталога. Для удаление <u>ПУСТОГО</u> каталога применяется: rd (дискковод:) <i>путь</i>.</p>	<p>Уничтожение каталога. Для удаления <u>ПУСТОГО</u> каталога применяется команда rmdir <i><имя директории></i>.</p>

Лабораторная работа №2

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

Цель работы: изучить формы представления чисел в различных системах счисления и правила выполнения арифметических операций над ними.

Задание и порядок выполнения лабораторной работы

1. В соответствии с вариантом выбрать из таблицы 4 три исходных числа A_1 , A_2 , A_3 в десятичной системе счисления.

2. Осуществить перевод числа $W=A_2,A_3$ (A_2 — целая часть числа W , A_3 — дробная), заданного в десятичной СС в системы счисления с основаниями 2, 8 и 16.

При переводе дробной части числа задается следующая точность представления:

- для двоичной СС — 8 разрядов после запятой
- для восьмеричной и шестнадцатеричной — 3 разряда после запятой.

3. Обратным переводом проверить правильность полученных результатов.

4. Выполнить перевод числа A_1 в системы счисления с основаниями 2, 8 и 16.

5. Обратным переводам проверить правильность полученных результатов.

6. Перевести числа $+(-)A_1$ $+(-)A_2$ $+(-)A_3$ (положительное и отрицательное) в прямой дополнительный код в двоичной СС.

7. Выполнить над числами A_2 и A_3 следующие операции в двоичной СС:

$$(A_2+A_3), (A_2-A_3), (-A_2+A_3), (-A_2-A_3).$$

Результат записать в прямом и дополнительном кодах. Провести проверку переводом в десятичную СС, и сравнением полученного результата с результатам вычислений в десятичной системе счисления. Выявить возможные случаи переполнения разрядной сетки.

8. Результаты вычислений перевести в восьмеричную и

шестнадцатеричную системы счисления используя триады и тетрады.

9. Представить число W , определенное в п.2 задания, в форме с плавающей запятой в десятичной, двоичной, восьмеричной и шестнадцатеричной системах счисления.

10. Результаты выполнения заданий привести в отчете по лабораторной работе.

Таблица 4. Варианты чисел

В а р и а н т	А	В	С
1	2 7 4	7 4	4 5
2	1 4 8	8 6	5 1
3	2 8 2	6 5	3 4
4	2 2 6	1 3	3 8
5	1 6 2	6 7	2 7
6	3 0 2	3 2	3 8
7	1 7 5	2 6	2 2
8	2 7 4	7 2	6 2
9	2 5 2	1 8	4 2
10	2 1 4	1 9	7 6
11	1 5 5	2 6	5 2
12	1 6 7	2 0	8 4
13	3 1 9	5 9	3 9
14	1 2 5	4 2	3 4
15	2 4 4	5 5	8 3
16	1 8 1	7 9	8 7
17	1 9 8	8 3	5 4
18	2 2 4	2 6	3 5
19	2 3 7	6 7	2 1
20	1 8 3	5 8	4 1
21	1 8 8	4 6	8 9
22	2 4 5	3 7	1 8
23	2 6 3	3 1	3 1
24	3 1 5	7 4	8 6
25	1 4 9	4 7	8 7

Содержание отчёта

1. Титульный лист.
2. Название и цель работы.
3. Результаты выполнения индивидуального задания:
 - а. оформляется каждый пункт задания;

- b. все выполняемые действия должны быть представлены;
 - c. произведена проверка в необходимых пунктах;
4. Выводы по работе соответственно цели лабораторной работы.

Приложение к лабораторной работе

Основные сведения о системах счисления

Позиционные системы счисления

Выполнение любых вычислений базируется на определенной форме представления чисел. Это определяется принятой **системой счисления** - совокупностью символов и правил для представления чисел. Символы называются цифрами данной системы счисления. Системы счисления могут быть *позиционными* и *непозиционными*.

Непозиционной системой называется такая, в которой значение символа не зависит от его места расположения в числе. Для образования числа в непозиционной системе счисления используются операции арифметического сложения и вычитания. Примером непозиционной системы счисления является римская:

$$XVI = X + V + I = 10 + 5 + 1 = 16_{10}$$

$$IX = X - I = 10 - 1 = 9_{10}$$

В *позиционной* системе счисления значение каждого символа-цифры зависит от его места расположения в числе. Справедливо следующее представление числа в позиционной системе счисления:

$$x(q) = a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \dots + a_1q^1 + a_0q^0 + a_{-1}q^{-1} + \dots + a_mq^m \quad (1)$$

где

$x(q)$ — число в системе счисления с основанием q ;

a_i — цифра i -ого разряда в числе;

n — количество целых разрядов числа;

m — количество дробных разрядов числа.

Под основанием системы счисления q , с одной стороны, понимают количество различных цифр, ее образующее, а с другой стороны - число, показывающее во сколько раз вес цифры данного разряда меньше веса

соседнего старшего разряда.

Очевидно, что используемая система счисления определяет набор правил (алгоритмов) выполнения операций над числами. Поэтому важное значение имеет правильное представление чисел и преобразование чисел в различных системах счисления.

Наибольшее распространение в вычислительной технике имеют системы счисления с основаниями 2, 8, 16 — двоичная, восьмеричная, шестнадцатеричная. При переводе числа в шестнадцатеричную систему счисления необходимо помнить, что шестнадцатеричные числа представляются символами 0, 1, 2, ..., 8, 9, A(10), B(11), C(12), D(13), E(14), F(15).

Перевод чисел из одной системы счисления в другую

Здесь различают три ситуации при переводе чисел:

- перевод числа из десятичной системы в систему с любым основанием;
- перевод числа из системы с любым основанием в десятичную;
- перевод числа из системы с основанием q_1 в систему с основанием q_2 .

Правила, используемые для перевода целых и дробных чисел различны.

Для перевода целого числа из десятичной системы счисления в систему с основанием q , число нужно последовательно делить на основание q до тех пор, пока не будет получена целая часть частного, равная 0, то есть будет получен остаток от деления, меньший q .

Число в системе счисления с основанием q записывается в виде упорядоченной последовательности остатков от деления в порядке, обратном получению остатков, то есть старшей цифрой числа будет последний остаток.

Для перевода правильной дроби из десятичной системы счисления в систему с основанием q число нужно последовательно умножать на основание q (причем умножению подвергаются только дробные части) до тех пор, пока не будет обеспечена заданная точность представления числа.

Дробь в системе счисления с основанием q записывается в виде упорядоченной последовательности целых частей произведений в порядке их получения. Если требуемая точность перевода q , то число указанных последовательных произведений (то есть цифр в представлении дроби) равно

$k+1$. По $(k+1)$ -ой цифре производится округление k -той цифры.

Если на некотором шаге получения произведений дробная часть числа становится равной 0, то процесс преобразования на этом заканчивается, так как все остальные цифры в представлении дроби будут равны 0.

При переводе неправильной дроби из десятичной системы счисления в систему с основанием q отдельно переводится целая и дробная части числа.

При переводе чисел из системы счисления с основанием q_1 в систему с основанием q_2 выполняется промежуточное преобразование в десятичную систему.

Связь двоичной, восьмеричной и шестнадцатеричной систем счисления.

Восьмеричная и шестнадцатеричная системы счисления относятся к двоично-кодированным системам, основание которых представляют целые степени двойки: 2^3 — для восьмеричной и 2^4 — для шестнадцатеричной.

Каждая восьмеричная цифра представляется триадой двоичных цифр, а каждая шестнадцатеричная цифра - тетрадой двоичных цифр.

Перевод целых и дробных чисел из двоичной в восьмеричную и из двоичной в шестнадцатеричную системы счисления производится с учетом следующей таблицы:

Таблица 5. Представление чисел

Число	Триада	Тетрада
0	000	0000
1	001	0001
2	010	0010
3	011	0011
4	100	0100
5	101	0101
6	110	0110
7	111	0111
$8(2^3)$	1 000	1000
9		1001
A(10)		1010

B(11)		1011
C(12)		1100
D(13)		1101
E(14)		1110
F(15)		1111
16 (2^4)		1 0000

Для перевода двоичного числа в восьмеричную (шестнадцатеричную) систему счисления число разбивается на триады (тетрады) двоичных цифр. Причем для целого числа триады (тетрады) находятся, начиная с младшего разряда, двигаясь влево к старшему разряду. Если старшая триада (тетрада) не получается из-за нехватки цифр, то слева к числу приписывается нужное количество нулей.

Для дробного числа триады (тетрады) находятся, начиная со старшего разряда, двигаясь вправо к младшему. Если количество разрядов не кратно трём (четырем), то справа приписывается нужное количество нулей. Далее каждой триаде (тетраде) ставится в соответствие восьмеричная (шестнадцатеричная) цифра.

При обратном переводе вместо каждой восьмеричной (шестнадцатеричной) цифры записывается эквивалентная ей триада (тетрада) двоичных. Положение запятой между целой и дробной частями числа сохраняется. Нули слева от целой части и справа от дробной части опускаются.

Формы представления чисел в ЦВМ

Общий вид представления чисел.

В памяти ЦВМ числовая информация может быть представлена в различных формах.

В случае с фиксированной запятой для всех чисел, над которыми выполняются операции, положение запятой строго зафиксировано между целой и дробной частями числа.

Обычно в ЦВМ используются два способа расположения запятой:

➤ перед старшим разрядом, то есть целая часть числа равна нулю, и в операциях участвуют правильные дроби;

➤ после младшего разряда, то есть дробная часть числа равна нулю, и в операциях участвуют только целые числа.

Число с фиксированной запятой представляется следующим образом:

$$[X]_{\text{ф.з.}} = X * K_m, \quad (2)$$

где $[X]_{\text{ф.з.}}$ — машинное представление числа с фиксированной запятой;

X — исходное число,

K_m — масштабный коэффициент, который выбирается из условий конкретной разрядной сетки и не должен допускать выхода исходных чисел и результатов вычислений за пределы допустимого диапазона.

Масштабный коэффициент должен быть единым для всех обрабатываемых в машине чисел и получаемых результатов, он хранится отдельно от представляемых чисел и учитывается при выдаче конечного результата.

Число в форме с фиксированной запятой должно удовлетворять следующему неравенству:

$$[X]_{\text{ф.з.}}.\text{min} \leq [X]_{\text{ф.з.}} \leq [X]_{\text{ф.з.}}.\text{max} \quad (3)$$

Если нарушена левая часть неравенства, то имеем машинный ноль; если нарушена правая часть неравенства, то произошло переполнение разрядной сетки.

Представление чисел в форме с плавающей запятой позволяет избежать масштабирования исходных чисел, а также увеличить диапазон и точность представляемых чисел.

Число в *нормальной форме* имеет вид:

$$X = m * q^p, \quad (4)$$

где q — основание СС,

p — целое число — порядок числа X ,

m — мантисса числа.

Поскольку, изменяя одновременно определённым образом мантиссу и порядок числа X , можно по выражению (4) получить любое количество представлений числа X , то на мантиссу m накладывается следующее ограничение, чтобы избежать неоднозначности в представлении чисел

$$q^{-1} \leq |m| \leq 1. \quad (5)$$

Если для числа X в форме с плавающей запятой выполнены условия (5), то

число X называется нормализованным, мантисса представляется правильной дробью, а ее старший разряд с основанием q отличен от 0. Операция преобразования ее к виду (5) называется нормализацией.

Чтобы нормализовать мантиссу, ее нужно сдвигать вправо для целого числа и влево для дроби на столько разрядов, чтобы целая часть мантиссы была равна нулю, а старший разряд мантиссы был равен 1, после чего к порядку целого числа прибавить (а из порядка дроби вычесть) столько единиц, на сколько разрядов был произведен сдвиг.

Разрядная сетка для числа с плавающей запятой состоит из двух частей: для порядка и для мантиссы.

<i>Порядок</i>				<i>Мантисса</i>			
	$m-1$	$m-2$	0	0	1	2	n
Знак	2^{m-1}	2^{m-2}	2^0	Знак	2^{-1}	2^{-2}	2^{-n}
порядка				мантиссы			

Кодирование отрицательных чисел.

Отрицательные числа в ЭВМ представляются в прямом, дополнительном и обратном кодах.

При представлении числа в прямом коде код числа в разрядной сетке из n разрядов совпадает с самим числом.

В двоичной системе счисления знаки кодируются следующим образом: «+» цифрой 0, «-» цифрой 1. Знаковый разряд располагается слева от значащих цифр числа, то есть это крайний левый разряд в представлении числа. При таком подходе введение знакового разряда уменьшает диапазон представления положительных и отрицательных чисел. Вес знакового разряда соответствует весу старшего разряда разрядной сетки. Значащие разряды числа занимают $n-1$ разряд.

Диапазон представления положительных чисел $0 \leq X \leq 2^{n-1} - 1$, а отрицательных $0 \geq X \geq -(2^{n-1} - 1)$.

$$[X]_{пр} = \begin{cases} X, & \text{при } X \geq 0 \\ q^{n-1} + |X|, & \text{при } X < 0 \end{cases} \quad (6)$$

где

q^{n-1} величина, равная весу старшего разряда (для дробных чисел $q^{n-1} = 1$).

Обратите внимание на то, что запись числа в форме (6) приводит к двум равнозначным формам записи нуля — «положительной» и «отрицательной».

Дополнительный код отрицательного числа представляет собой дополнение модуля исходного числа X до q^n , где n - количество разрядов целой части числа X . Для правильной дроби $q^n = 10$ в соответствующей системе счисления.

Поэтому для целого числа:

$$[X]_{\text{доп}} = \begin{cases} X, & \text{при } X \geq 0 \\ q^{n-1} - |X|, & \text{при } X < 0 \end{cases} \quad (7)$$

Для правильной дроби:

$$[x]_{\text{доп}} = \begin{cases} X, & \text{при } X \geq 0 \\ 1 - |X|, & \text{при } X < 0 \end{cases} \quad (8)$$

Дополнительный код очень удобен при сложении чисел со знаком и вычитании. Действительно, если для трёхразрядного положительного числа \mathbf{B} в любой системе счисления дополнение равно $\mathbf{B}_{\text{доп}} = 1000 - |\mathbf{B}|$, то разность

$$\mathbf{A} - \mathbf{B} = \mathbf{A} + \{1000 - |\mathbf{B}|\} - 1000 = \mathbf{A} + \mathbf{B}_{\text{доп}} - 1000.$$

Таким образом, вычитание сводится к сложению с дополнением вычитаемого. Единица, возникающая за пределами разрядной сетки за счёт вычитания 1000 должна быть отброшена.

Старший разряд двоичного числа в дополнительном коде имеет смысл знакового разряда. Его нулевое значение соответствует положительному числу, а единичное — отрицательному.

Перевод в дополнительный код для двоичной системы счисления может быть выполнен по следующему правилу: все разряды двоичного числа, включая знаковый, инвертируются и к результату прибавляется единица младшего разряда.

Обратный код отрицательных чисел представляется как дополнение модуля исходного целого числа X до q^{n+1} и модуля исходной правильной дроби до 1 в соответствующей СС без единицы младшего разряда числа.

Если к представлению исходного числа в обратном коде прибавить единицу младшего разряда, то получим исходное число в дополнительном коде.

Сложение чисел в форме с фиксированной запятой.

Сложение чисел в прямом коде очень просто выполняется для чисел с одинаковыми знаками. Числа складываются, а их сумме присваивается код знака слагаемых.

Если числа имеют разные знаки, то используется операция алгебраического сложения (вычитания). Из большего по модулю числа вычитается меньшее по модулю число, их разности присваивается код знака большего по модулю числа.

Гораздо проще в этом случае использовать дополнительный код для отрицательных чисел. Тогда операция алгебраического сложения сводится к простому арифметическому сложению. Знаковый разряд и цифровая часть числа рассматриваются как единое целое. Машина оперирует с отрицательными числами, представленными в дополнительном коде, как с положительными независимо от того, целые они или дробные. Сумма также получается в дополнительном коде. Правильный знак суммы получается автоматически в процессе суммирования знаковых цифр операндов и цифры переноса из соседнего младшего разряда.

В случае возникновения единицы переноса из знакового разряда суммы ее нужно отбросить ("погасить") при сложении в дополнительном коде; прибавить к младшему разряду суммы при сложении в обратном коде (циклический перенос единицы переполнения).

Суммирование двоичных чисел происходит в соответствии с таблицей сложения, причем в каждом i -том разряде участвуют в сложении три цифры: цифра переноса из соседнего младшего разряда P_i и две цифры соответствующего разряда A_i и B_i . В результате сложения имеем сумму S_i в данном i -том разряде и перенос в следующий $(i+1)$ -ый разряд P_{i+1} .

Таблица 3.

P_i	A_i	B_i	S_i	P_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0

1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Формирование признака переполнения разрядной сетки

При алгебраическом сложении двух чисел, помещающихся в разрядную сетку, может возникнуть переполнение, то есть образоваться сумма, для представления которой разрядная сетка должна содержать на один разряд больше, чем разрядная сетка для представления слагаемых.

Признак обнаружения переполнения разрядной сетки формируется следующим образом: если при сложении двух чисел в дополнительном (обратном) коде существует перенос в знаковый разряд суммы при отсутствии переноса из её знакового разряда, или же существует перенос из знакового разряда суммы при отсутствии переноса в её знаковый разряд, то фиксируется ситуация переполнения разрядной сетки, причем в первом случае — положительное переполнение, а во втором — отрицательное.

Если и в знаковый и из знакового разряда суммы перенос существует или его нет, то переполнение отсутствует.

При положительном переполнении результат операции положительный, а при отрицательном - отрицательный.

Пример: сложить следующие пары чисел в дополнительном коде:

- | | | | |
|----|---------------|----|---------------|
| а) | $X1 = +01010$ | б) | $X1 = -01101$ |
| | $X2 = +01011$ | | $X2 = -01001$ |

а)	$[X1]_{\text{доп}} = 01010$
	$[X2]_{\text{доп}} = 01011$

$$[X_{\text{сум}}]_{\text{доп}} = 10101$$

Результат сложения двух положительных чисел получился с отрицательным знаком, что невозможно. Но при сложении существовал перенос в знаковый разряд суммы при отсутствии его из знакового разряда. Это ситуация положительного переполнения разрядной сетки. Единица,

находящаяся в знаковом разряде — это старший разряд суммы, знак которой должен быть положительным.

Ответ: $[X_{\text{сум}}]_{\text{доп}} = [X_{\text{сум}}]_{\text{пр}} = 010101$ или $X_{\text{сум}} = +10101$

б) $[X1]_{\text{доп}} = 10011$

$[X2]_{\text{доп}} = 10111$

$[X_{\text{сум}}]_{\text{доп}} = 101010$

По правилу сложения в дополнительном коде единица переполнения из знакового разряда должна быть погашена, но результат сложения двух отрицательных чисел получится положительным, что невозможно. В этом случае существовал перенос из знакового разряда при отсутствии переноса в знаковый. Имеет место отрицательное переполнение разрядной сетки. Единица переполнения из знакового разряда - это отрицательный знак результата, а ноль в знаковом разряде — старший разряд суммы в дополнительном коде.

Поэтому: $[X]_{\text{доп}} = \underline{1}01010$; $[X2]_{\text{пр}} = \underline{1}10110$

Библиографический список

1. Андреева Е.В. Системы счисления и компьютерная арифметика: Учеб. пособие.- М.:БИНОМ. Лаборатория знаний,2004.-254 с.
2. Березин Б.И., Березин С.Б. Начальный курс С и С++. М.: Диалог-МИФИ, 2000. 288 с.
3. Ваулина Е.Ю., Рычков В.Н. Термины современной информатики: программирование, вычислительная техника, Интернет: Англо-русский, русско-английский словарь. – М.: Изд-во Эксмо, 2005.-640 с.
4. Каймин В.А., Питеркин В.М., Уртминцев А.Г. Информатика. - М.: БРИДЖ, 1994. – 208 с.
5. Касаткин А.И., Вальвачев А.Н. От Turbo С к Borland С++: Справочное пособие. Минск: Вышэйшая школа, 1992. 240 с.
6. Кортмен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ, второе издание: пер. с англ. – М.: Издательский дом «Вильямс», 2007. – 1296 с.
7. Левин А.Ш. Самоучитель работы на компьютере.9-е изд.- СПб.:Питер,2006.-748 с.
8. Лозовский В.Н. Информация, информатика, реальность. Введение в информатику: Учеб. пособие. – СПб.: Лань,2008.- 96 с.
9. Лютикас В.С. Факультативный курс по математике. Теория вероятностей. М.: Просвещение, 1990.
- 10.Могилев А.В. Информатика: Учеб. пособие.- М.,2004.- 825 с.
- 11.Нарышкин А.К. Цифровые устройства и микропроцессоры: Учеб. пособие для студ.высш.учеб.заведений. – М.: Издательский центр «Академия»,2006.-320 с.
- 12.Платонов Ю. М. и др. Электронный экзаменатор у вас дома. Информатика. - М.,2004.-176 с. (Серия «Библиотека студента»).
- 13.Прокушев Л.А. Сравнительное изучение языков Си и Паскаль: Учебно-справочное пособие. С-Пб.: СПбГААП, 1997. 152 с.
- 14.Рыжих Д.Н. Сборник задач, тестов и контрольных вопросов по дисциплине «Информатика».- СПб,2006.- 106с.

- 15.Симонович С.В. Информатика: Базовый курс. – СПб.: Питер , 2004.-640 с.
16.Сырецкий Г.А. Информатика. Фундаментальный курс.- СПб.: БХВ, 2005.

Оглавление

ВЕДЕНИЕ.....	3
Контрольная работа №1.....	4
ОПРЕДЕЛЕНИЕ КОЛИЧЕСТВА ИНФОРМАЦИИ, СОДЕРЖАЩЕГОСЯ В СООБЩЕНИИ.....	4
Порядок выполнения лабораторной работы.....	4
Содержание отчёта.....	5
Приложение к лабораторной работе.....	6
Лабораторная работа №1.....	16
РАБОТА В КОМАНДНОЙ СТРОКЕ.....	16
Задание и порядок выполнения лабораторной работы.....	16
Приложение к лабораторной работе.....	17
Лабораторная работа №2.....	30
ПРЕДСТАВЛЕНИЕ ЧИСЕЛ.....	30
Задание и порядок выполнения лабораторной работы.....	30
Содержание отчёта.....	31
Приложение к лабораторной работе.....	32
Библиографический список.....	42
Оглавление.....	43