

Министерство науки и высшего образования Российской Федерации  
ФГБОУ ВО «Кубанский государственный технологический университет»  
Кафедра Информационных систем и программирования

### **Информационные технологии**

Методические указания по выполнению лабораторных работ для студентов  
всех форм обучения направления 09.03.04 Программная инженерия

Краснодар  
2019

Составитель ст. преп. А. А. Ковтун

УДК 004.4

**Информационные технологии:** методические указания по выполнению лабораторных работ для студентов всех форм обучения направления 09.03.04 Программная инженерия. / Сост.: А. А. Ковтун; Кубан. гос. технол. ун-т. Каф. Информационных систем и программирования, 2019 – 41 с.

Составлены в соответствии с рабочей программой курса «Информационные технологии» для студентов всех форм обучения направления 09.03.04 Программная инженерия.

Изложены темы и варианты заданий для выполнения лабораторных работ.

## Содержание

1. Эффективное кодирование информации методом Шеннона-Фано. ....	4
2. Эффективное кодирование информации методом Хаффмана. ....	6
3. Исследование алгоритмов архивации (текстовые файлы). ....	8
4. Использование текстового процессора. Шрифтовое оформление, форматирование абзацев, таблиц, списков. ....	10
5. Использование текстового процессора. Создание структуры сложных документов, Создание сложных документов, автооглавление. ....	15
6. Табличный процессор. Создание электронных таблиц, форматирование, выполнение вычислительных расчётов. Использование абсолютных, относительных и смешанных ссылок. ....	21
7. Использование сложных функций табличного процессора. Сводные таблицы, диаграммы, макросы. ....	26
8. Создание мультимедийных презентаций. Создание структуры, формирование слайдов, управление показом презентации. ....	29
9. Создание и редактирование цифрового изображения. Параметры, слои, инструменты. Использование фильтров и эффектов. ....	32
10. Создание аудио и видео файлов. Нелинейный монтаж, наложение эффектов перекодирование формата. ....	34
11. Антивирусная защита. Использование антивирусного ПО. ....	34
12. Практическое применение криптографии. Шифр Вернама. ....	37

## 1. Эффективное кодирование информации методом Шеннона-Фано.

### 1 Цель работы

Научиться производить эффективное кодирование информации методом Шеннона-Фано. Разработать приложение для построения эффективного кода.

### 2 Краткая теория

Построение эффективного кода по методу Шеннона-Фано сводится к следующей процедуре:

- множество сообщений располагают в порядке убывания вероятностей;
- первоначальный ансамбль кодируемых сигналов разбивают на две группы таким образом, чтобы суммарные вероятности сообщений обеих групп были по возможности равны;
- одной группе присваивается символ 0, другой группе – символ 1;
- каждую из подгрупп делят на две группы так, чтобы их суммарные вероятности были по возможности равны;
- одним подгруппам каждой из групп вновь присваивают 0, а другим - 1, в результате чего получают вторые цифры кода. Затем каждую из четырех подгрупп вновь делят на равные (с точки зрения суммарной вероятности) части и т.д. до тех пор, пока в каждой из подгрупп остается по одной букве.

В качестве примера построим код Шеннона - Фано, в котором вероятности появления букв подчиняются закону, представленному в таблице, 1.

Т а б л и ц а 1 - Код Шеннона - Фано

Буква	Р	Кодовая комбинация
a1	1/4	11
a2	1/4	10
a3	1/8	011
a4	1/8	010
a5	1/16	0011
a6	1/16	0010
a7	1/16	0001
a8	1/16	0000

### 3 Задание

1. Составить программу в соответствии с вариантом задания для построения эффективного кода по методу Шеннона-Фано.

2. Провести эффективное кодирование ансамбля из  $N$  знаков по методу Шеннона - Фано.

#### 4 Варианты заданий

Сумма вероятностей появления символов равна единице. Вероятность каждого знака генерируется случайным образом. Число знаков  $N$  смотреть в таблице в соответствии с вариантом.

Вариант	1	2	3	4	5	6	7	8	9	10
N	18	20	24	26	22	27	21	19	23	25

## 2. Эффективное кодирование информации методом Хаффмана.

### 1 Цель работы

Научиться производить эффективное кодирование информации методом Хаффмана. Разработать приложение для построения эффективного кода.

### 2 Краткая теория

метод кодирования Шеннона-Фано не всегда приводит к эффективному коду. Поэтому на практике широко применяется методика построения ОНК с помощью кодовых деревьев, впервые описанная Хаффманом (рис. 2).

Для построения кода Хаффмана необходимо соединять попарно кодируемые символы с наименьшими вероятностями с учетом вновь образованных символов, имеющих суммарную вероятность предшествующей пары, пока вероятность очередной пары не станет равной 1. Если каждой верхней ветви, соединяющей символы, присваивать значение 0, а нижней 1, то образуется кодовое дерево, представляющее код.

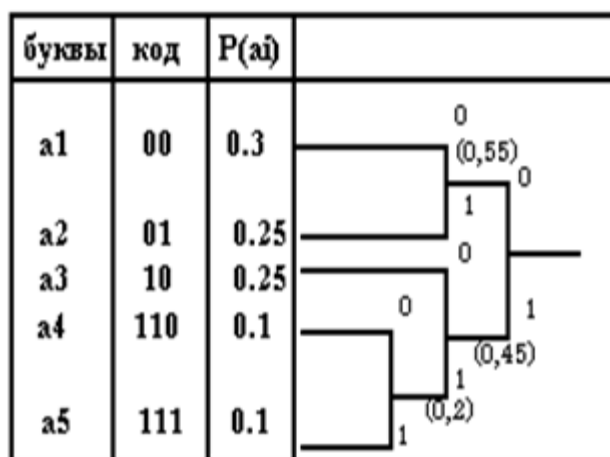


Рисунок 2

Преимущество метода Хаффмана сказывается при построении ОНК для вторичных алфавитов с  $m > 2$ . Большая точность достигается за счет более строгого выбора числа наименее вероятных символов первичного алфавита, объединяемых на первом этапе построения кодового дерева.

Преимущество оптимальных кодов: при прочих равных условиях оптимальные коды позволяют вести передачу информации с максимальной скоростью.

Недостаток оптимальных кодов: при прочих равных условиях оптимальные коды наиболее беззащитны от влияния помех, т.е. обладают наименьшей помехоустойчивостью

### 3 Задание

1. Составить программу в соответствии с вариантом задания для построения эффективного кода по методу Хаффмана.
2. Провести эффективное кодирование ансамбля из N знаков по методу Хаффмана.

### 4 Варианты заданий

Сумма вероятностей появления символов равна единице. Вероятность каждого знака генерируется случайным образом. Число знаков N смотреть в таблице в соответствии с вариантом.

Вариант	1	2	3	4	5	6	7	8	9	10
N	10	12	14	11	13	15	9	16	17	18

### 3. Исследование алгоритмов архивации (текстовые файлы).

#### 1 Цель работы

Практическое знакомство с алгоритмами сжатия информации. Реализация базового алгоритма RLE.

#### 2 Краткая теория

Алгоритм RLE является самым быстрым, простым и понятным алгоритмом сжатия данных и при этом иногда оказывается весьма эффективным. Именно подобный алгоритм используется для сжатия изображений в файлах *PCX*.

Он заключается в следующем: любой последовательности повторяющихся входных символов ставится в соответствие набор из трех выходных символов: первый-байт префикса, говорящий о том, что встретилась входная повторяющаяся последовательность, второй-байт, определяющий длину входной последовательности, третий-сам входной символ-*<prefix, length, symbol>*. Лучше всего работу алгоритма пояснить на конкретном примере.

Например: пусть имеется (шестнадцатиричный) текст из 20 байт:

05 05 05 05 05 05 01 01 03 03 03 03 03 03 05 03 FF FF FF FF

Выберем в качестве префикса байт FF. Тогда на выходе архиватора мы получим последовательность:

FF 06 05 FF 02 01 FF 06 03 FF 01 05 FF 01 03 FF 04 FF

Ее длина-18 байт, то есть достигнуто некоторое сжатие. Однако, нетрудно заметить, что при кодировании некоторых символов размер выходного кода возрастает (например, вместо 01 01 - FF 02 01). Очевидно, одиночные или дважды (трижды) повторяющиеся символы кодировать не имеет смысла - их надо записывать в явном виде. Получим новую последовательность:

FF 06 05 01 01 FF 06 03 05 03 FF 04 FF

длиной 13 байт. Достигнутая степень сжатия:  $13/20 \cdot 100 = 65\%$ .

Нетрудно заметить, что префикс (маркер) может совпасть с одним из входных символов. В этом случае входной символ может быть заменен своим “префиксным” представлением, например:

FF то же самое, что и FF 01 FF (три байта вместо одного).



Поэтому, от правильного выбора префикса зависит качество самого алгоритма сжатия, так как, если бы в нашем исходном тексте часто встречались одиночные символы FF, размер выходного текста мог бы даже превысить входной. В общем, случае в качестве префикса следует выбирать самый редкий символ входного алфавита.

Таким образом, при выполнении лабораторной работы на первом этапе необходимо найти самый редко встречаемый в кодируемом файле символ, для того, чтобы использовать его в качестве маркера. Следует отметить, что в качестве маркера может быть использован любой символ, который ни разу не встретился в файле.

### **3 Задание**

Написать программу архиватор:

1. Написать функции чтения и записи данных в двоичный файл.
2. Реализовать алгоритм поиска префикса в байтовом массиве.
3. Реализовать алгоритм кодирования данных посредством RLE кода.
4. Предложить и реализовать алгоритм декодирования RLE кода.

### **4 Варианты заданий**

Для сжатия информации использовать текстовый файл, состоящий из 100-N символов. N – соответствует номеру варианта по списку.

#### 4. Использование текстового процессора. Шрифтовое оформление, форматирование абзацев, таблиц, списков.

##### 1 Цель работы

Изучить основные приемы работы в текстовом процессоре; элементы форматирования текста, создания и использования стилей; форматирование абзацев текста. Рассмотреть принципы построения таблиц, форматирование, создание сложных документов.

##### 2 Ход выполнения работы:

###### Часть 1

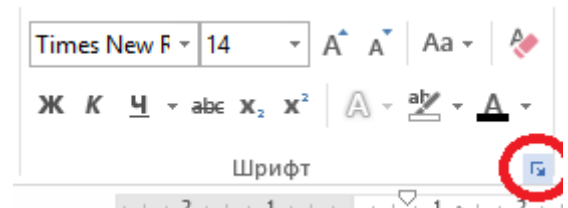
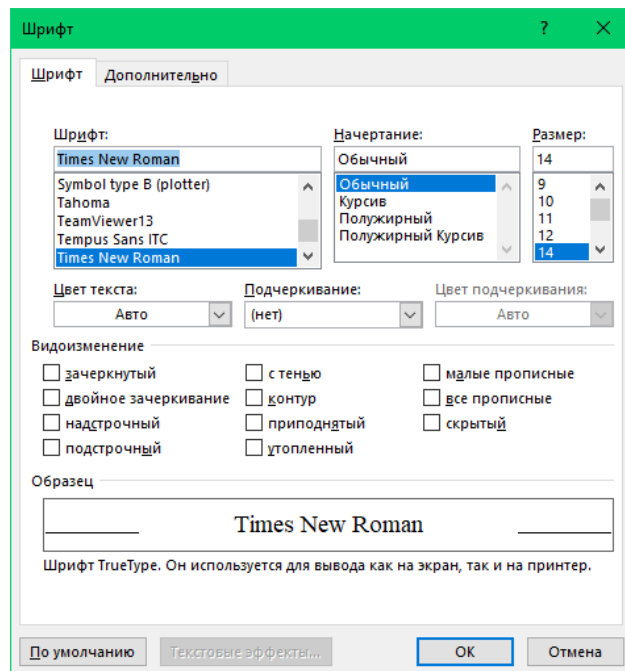
1. Создайте новый текстовый документ Libre Writer
2. Напечатайте текст (распечатанный или электронный документ)
3. Отработайте различные режимы выделения текста, запишите таблицы в тетрадь:

Комбинация	Описание
Ctrl+A	Выделить весь текст
Shift+ →	Выделить текст с текущей позиции курсора по символу вправо
Shift+ ↓	Выделить текст с текущей позиции курсора по строкам вниз
Shift+Home	Выделить текст с текущей позиции курсора до начала строки
Shift+End	Выделить текст с текущей позиции курсора до конца строки
Ctrl+Shift+ Home	Выделить текст с текущей позиции курсора до конца документа
Ctrl+Shift+ End	Выделить текст с текущей позиции курсора до начала документа
Двойное нажатие ЛКМ (левой кнопкой мыши)	Выделить слово
Тройное нажатие ЛКМ	Выделить абзац

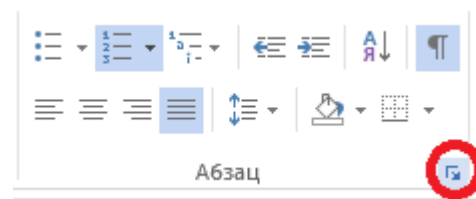
Комбинация	Описание
Ctrl+E	Выравнивание по центру
Ctrl+J	Выравнивание по ширине

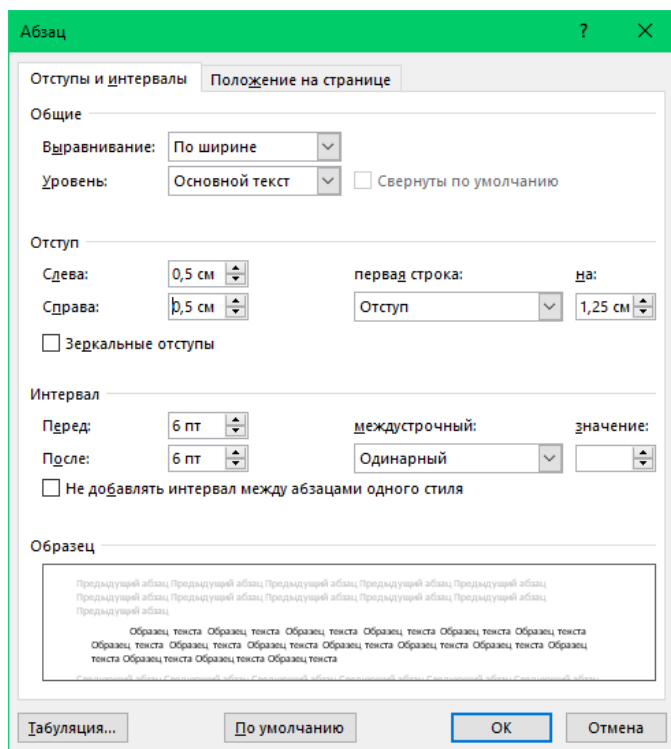
Ctrl+L	Выравнивание по левому краю
Ctrl+R	Выравнивание по правому краю
Ctrl+I	Курсивное начертание
Ctrl+B	Полужирное начертание
Ctrl+U	Подчеркнутое начертание

4. Отформатируйте **выравнивания и начертания** по образцу используя окно настроек шрифта и комбинации горячих клавиш:



5. Отформатируйте абзацы, используя окно настроек:





6. Изучите основные непечатаемые символы, они помогут ориентироваться в структуре текста:

Символ	Значение
¶	Новый абзац (Enter)
□	Ячейка таблицы
↵	Переход на новую строку без разрыва абзаца (Shift+Enter)
·	Пробел
◦	Короткий пробел (Ctrl+Shift+пробел)
.....разрыв.....	Различные разрывы (Разметка страницы → Разрывы)

Вставьте разрыв страницы перед текстом и на первой оформите ручной титульный лист по образцу, не используя ручные отступы:

## Часть 2


1. Создайте новый текстовый документ Libre Writer
2. Оформите титульную страницу (вкладка вставка)
3. Вставьте таблицу размером 2x3, в каждую ячейку запишите четверостишие гимна РФ:

Россия — священная наша держава, Россия — любимая наша страна. Могучая воля, великая слава — Твоё достоянье на все времена! Славься, Отечество наше свободное, Братских народов союз вековой, Предками данная мудрость народная! Славься, страна! Мы гордимся тобой! От южных морей до полярного края Раскинулись наши леса и поля. Одна ты на свете! Одна ты такая — Хранимая Богом родная земля! Славься, Отечество наше свободное, Братских народов союз вековой, Предками данная мудрость народная! Славься, страна! Мы гордимся тобой! Широкий простор для мечты и для жизни Грядущие нам открывают года. Нам силу даёт наша верность Отчизне. Так было, так есть и так будет всегда! Славься, Отечество наше свободное, Братских народов союз вековой, Предками данная мудрость народная! Славься, страна! Мы гордимся тобой!

4. Вставьте таблицу размерностью 2x2. Оформите в ней дату и место для подписи. Уберите границы таблицы:

19.02.2018	
	(подпись)

19.02.2018	
	(подпись)

5. Создайте в текстовом редакторе Microsoft Word таблицу из 5 строк и 4 столбцов: Вставка-Таблица/Вставить таблицу, в диалоговом окне указать число строк и столбцов. Выделите всю таблицу и установить шрифт Arial в 10 пунктов. Поставьте курсор в любое место таблицы и выполнить команды Макет/Выделить/Выделить таблицу. Поместите указатель мыши в верхний левый угол таблицы так, чтобы появился символ . Щелкнуть по нему левой кнопкой мыши. Дайте таблице заголовок, объединив ячейки первой строки: выделите строку и выполните команду Макет/Объединить ячейки (или команду контекстного меню (ПКМ) Объединить ячейки).
6. Заполните таблицу данными по образцу:

Свойства некоторых веществ			
Свойства	Кислород	Железо	Вода
Агрегатное состояние	Газ	Твердое	Жидкость
Запах	—	—	—
Цвет	—	Серое	—

7. Создайте таблицу. Разделите ячейки созданной Вами таблицы по приведенному образцу, используя команду Макет/Разбить ячейки (или через КМ). Аналогично образцу обамрите ячейки таблицы (Конструктор/Границы/Границы и заливка или через КМ):



11. Для подсчета суммы используется формула SUM (Вкладка Макет, Формула). Адресация ячеек в MS Word происходит также как и в MS Excel: первая ячейка таблицы носит имя A1, Вторая ячейка в первой строке – B1 и так далее по порядку. Набор ячеек можно указать в виде диапазона, например (A1:A8), либо используя ключевые слова:

Above – все ячейки над той, в которую записывается формула

Below - все ячейки под той, в которую записывается формула

Left – все ячейки слева

Right – все ячейки справа соответственно.

### **3 Варианты**

Варианты заданий отличаются тематикой набираемого текста:

Вариант	Тема
1	Экономика
2	Бухгалтерия
3	Билетная касса
4	Библиотека
5	Фильмотека
6	Погода
7	Автомобили
8	Велосипеды
9	Обучение
10	Животные

## **5. Использование текстового процессора. Создание структуры сложных документов, Создание сложных документов, автооглавление.**

### **1 Цель работы**

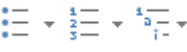

Изучение возможностей текстового процессора при работе с объектами: списки, таблицы, формулы, диаграммы, освоение приемов работы с объектами в текстовом документе. Изучение возможностей текстового процессора при работе с объектами: многоуровневые списки, колонки, создание автоматического оглавления, освоение приемов работы с объектами в текстовом документе.

### **2 Ход выполнения работы**


## Часть 1

1. Создайте таблицу по образцу:

Маркированный	Нумерованный	Многоуровневый
<ul style="list-style-type: none"> <li>○ Компьютерное оборудование</li> <li>○ Системный блок</li> <li>○ Монитор</li> <li>○ Клавиатура</li> <li>○ Принтер</li> <li>○ Программное обеспечение</li> <li>○ Операционный системы</li> <li>○ Прикладные программы</li> <li>○ Информационные материалы и документы</li> </ul>	<ol style="list-style-type: none"> <li>1. Компьютерное оборудование</li> <li>2. Системный блок</li> <li>3. Монитор</li> <li>4. Клавиатура</li> <li>5. Принтер</li> <li>6. Программное обеспечение</li> <li>7. Операционный системы</li> <li>8. Прикладные программы</li> <li>9. Информационные материалы и документы</li> </ol>	<ul style="list-style-type: none"> <li>✓ Компьютерное оборудование <ul style="list-style-type: none"> <li>• Системный блок</li> <li>• Монитор</li> <li>• Клавиатура</li> <li>• Принтер</li> </ul> </li> <li>✓ Программное обеспечение <ul style="list-style-type: none"> <li>• Операционный системы</li> <li>• Прикладные программы</li> </ul> </li> <li>✓ Информационные материалы и документы</li> </ul>

Для оформления списков используйте соответствующие инструменты на вкладке Главная в разделе Абзац . Для перехода на следующий уровень и обратно (для многоуровневого списка) используйте кнопки уменьшить\увеличить отступ .

2. Составьте таблицу формул (Вставка-Формула, выбор нужной

заготовки, например: верхний индекс , только потом заполнение конкретными значениями) по образцу:

Вариант 1		Вариант 2	
$E = \frac{1}{4\pi\epsilon_0 r^2} \sqrt{1+3\cos^2\alpha}$	1.	$\Psi = 4\pi \sum_{i=1}^n Q_i$	1.
$\Psi = \sum_{i=1}^n Q_i$	2.	$C = \frac{2\pi\epsilon_0 l}{\ln\left(\frac{r_2}{r_1}\right)}$	2.
$C = \frac{4\pi\epsilon_0 R_1 R_2}{R_2 - R_1}$	3.	$C = \frac{\epsilon R_1 R_2}{R_2 - R_1}$	3.
$\oint_l H_l dl = \frac{\delta\Phi_0}{\delta}$	4.	$\oint_l H_l dl = \frac{-\delta\Phi_l}{\delta}$	4.
$C = \frac{2\pi\epsilon_0 l}{\ln\left(\frac{r_2}{r_1}\right)}$	5.	$\Psi = \sum_{i=1}^n Q_i$	5.

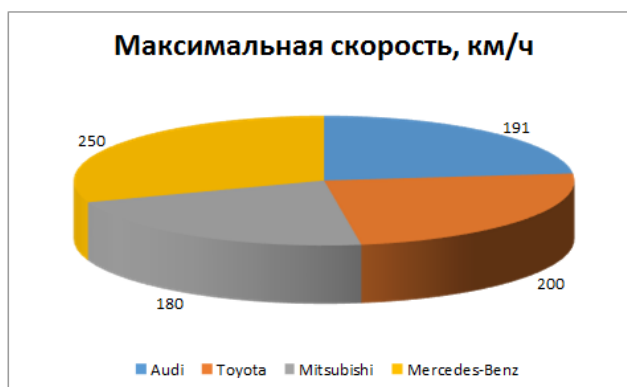
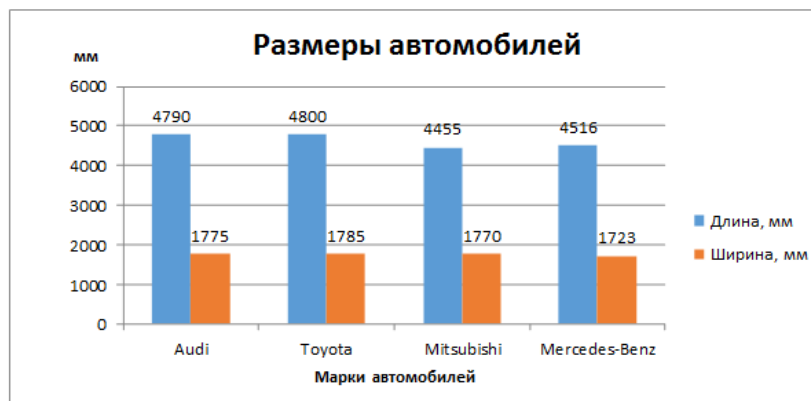
Один вариант на выбор, размер формул – 26.

3. Создайте диаграммы (Вкладка вставка – раздел Текст – объект – Диаграммы Microsoft Excel) по данным таблицы:



Наименование автомобиля	Длина, мм	Ширина, мм	Разгон, км/ч	Максимальная скорость, км/ч
Audi	4790	1775	100	191
Toyota	4800	1785	100	200
Mitsubishi	4455	1770	100	180
Mercedes-Benz	4516	1723	100	250

Оформите диаграммы по образцу:



## Часть 2

### 1. Составьте многоуровневый список по образцу:

Аппаратное обеспечение персонального компьютера

#### 1. Системный блок

##### а. Материнская плата

- і. Многослойная печатная плата, на которой размещены и соединены печатными проводниками различные компоненты электроники

1. ПЗУ
2. Контроллеры
3. Резисторы
4. ОЗУ

##### б. Видеокарта

- c. Сетевая карта
  - d. Процессор
  - e. Южный мост
  - f. Северный мост
- 2. Монитор
- 3. Устройства ввода
  - a. Манипуляторы
  - b. Сканеры
  - c. Цифровые камеры
  - d. Клавиатуры
    - i. Тип механизма
      - 1. Мембранный механизм
      - 2. Ножничный механизм
      - 3. Механический тип
- 4. Устройства вывода
  - a. Печатающие устройства
    - i. Принтеры
      - 1. Струйные
      - 2. Лазерные
      - 3. Матричные
      - 4. 3D
  - b. Устройства отображения информации
  - c. Устройства формирования объемных изображений
- 5. Накопители информации
  - a. На жестких магнитных дисках
    - i. Винчестеры
    - ii. RAID-системы
  - b. На твердотельной памяти
    - i. Модули flash-памяти
    - ii. Модули оперативной памяти
    - iii. Модули графической памяти
    - iv. SSD-накопители
  - c. На компакт-дисках
    - i. CD
    - ii. DVD

## 2. Составьте колонки по образцу:

Заметался пожар голубой,  
Позабылись родимые дали.  
В первый раз я запел про любовь,  
В первый раз отрекаюсь  
скандалить.

Был я весь как запущенный сад,  
Был на женщин и зелие падкий.  
Разонравилось пить и плясать  
И терять свою жизнь без оглядки.

Мне бы только смотреть на тебя,  
Видеть глаз златокарий омут,  
И чтоб, прошлое не любя,  
Ты уйти не смогла к другому.

Поступь нежная, легкий стан,  
Если б знала ты сердцем упорным,  
Как умеет любить хулиган,  
Как умеет он быть покорным.

Я б навеки забыл кабаки  
И стихи бы писать забросил,  
Только б тонко касаться руки  
И волос твоих цветом в осень.

Я б навеки пошел за тобой  
Хоть в свои, хоть в чужие дали...  
В первый раз я запел про любовь,  
В первый раз отрекаюсь  
скандалить.

### 3. Составьте автособираемое оглавление по образцу:

Программное обеспечение

Прикладное

Общего назначения

*Текстовые процессоры*

*Табличные процессоры*

Специального назначения

Профессионального уровня

Системное

Антивирусы

*Eset NOD32*

Архиваторы

*WinRAR*

Утилиты

Системы программирования

Язык программирования

Среда программирования

Оглавление

<u>Программное обеспечение</u> .....	20
<u>Прикладное</u> .....	20
<u>Общего назначения</u> .....	20
<u>Текстовые процессоры</u> .....	20
<u>Табличные процессоры</u> .....	20
<u>Специального назначения</u> .....	20
<u>Профессионального уровня</u> .....	20
<u>Системное</u> .....	20
<u>Антивирусы</u> .....	20
<u>Eset NOD32</u> .....	20
<u>Архиваторы</u> .....	20
<u>WinRAR</u> .....	20
<u>Утилиты</u> .....	20
<u>Системы программирования</u> .....	20
<u>Язык программирования</u> .....	20
<u>Среда программирования</u> .....	20

## 4 Варианты

Варианты заданий отличаются тематикой набираемого текста:

Вариант	Тема
1	Экономика
2	Бухгалтерия
3	Билетная касса
4	Библиотека
5	Фильмотека
6	Погода
7	Автомобили
8	Велосипеды
9	Обучение
10	Животные

## 6. Табличный процессор. Создание электронных таблиц, форматирование, выполнение вычислительных расчётов. Использование абсолютных, относительных и смешанных ссылок.

### 1 Цель работы

Изучение возможностей табличного процессора. Режимы работы с таблицей. Использование маркера автозаполнения для эффективного ввода данных. Выполнение расчётов по формулам, использование функций и стандартных процедур для анализа информации в таблице.

### 2 Порядок выполнения работы

#### Задание 1

Используя информацию из таблицы на листе 1 выполните автозаполнение

Начальное значение	Продолжение ряда
--------------------	------------------

1, 2, 3...	4, 5, 6...
09:00	10:00, 11:00, 12:00...
Пн	Вт, Ср, Чт...
Понедельник	Вторник, Среда, Четверг...
янв	фев, мар, апр...
янв, апр	июл, окт, янв...
янв-07, апр-07	июл-07, окт-07, янв-08...
15-янв, 15-апр	15-июл, 15-окт...

## Начальное значение Продолжение ряда

2018, 2019

2020, 2021, 2022...

1-январь, 1-март

1-май, 1-июль, 1-сентябрь...

кв. 3 (или квартал 3)

кв. 4, кв. 1, кв. 2...

текст1, текстА

текст2, текстА, текст3, текстА...

1-й период

2-й период, 3-й период...

Товар 1

Товар 2, Товар 3...

1. На Листе2 оформите таблицу по образцу. Переименуйте Лист2 в Продажа продукции. Для всех изменений следует предварительно выделить диапазон ячеек, вызвать по ним контекстное меню и выставить изменения на соответствующих вкладках

	A	B	C	D	E	F	G	H	I
1	Наименование товара	Цена закупки, руб.	Количество	Вес, кг		Транспортные расходы		Торговая надбавка	Цена продажи
2				1 ед.	всего	на 1 ед.	всего		
3								22,00%	
4	Товар 1	100,00р.	10	2		24,00р.			
5	Товар 2	200,00р.	2	3		40,00р.			
6	Товар 3	120,00р.	13	1,5		100,00р.			
7	Товар 4	130,00р.	5	2,1		53,00р.			
8	Товар 5	320,00р.	23	2		27,00р.			
9	Товар 6	122,00р.	11	0,5		16,00р.			
10	Товар 7	150,00р.	4	3,2		56,00р.			
11	Товар 8	120,00р.	3	1		22,00р.			
12									

Рекомендации по выполнению задания:

Использовать маркер автозаполнения для ввода данных в ячейки A4:A11;

Установить следующие форматы отображения данных:

- процентный для ячейки H3, денежный для данных в столбцах B, F;
- залить ячейки первых двух строк любым цветом;
- для ячейки H3 использовать другой цвет;
- для шапки таблицы использовать функцию переноса по словам;
- при необходимости выполнить объединение ячеек;
- границы таблицы и выравнивание данных в ячейках оформить по образцу;
- для подсчёта ячеек H4:H11 используйте x

1	9:00	пн	Понедельник	январь	январь	январь.07	15.январь	2018	01.январь	кв. 3	текст 1	1-й период	Товар 1
2	10:00	вт	Вторник	февраль	апрель	апрель.07	15.апрель	2019	01.март	кв. 4	Текст А	2-й период	Товар 2
3	11:00	ср	Среда	март	июль	июль.07	15.июль	2020	01.май	кв. 1	текст 2	3-й период	Товар 3
4	12:00	чт	Четверг	апрель	октябрь	октябрь.07	15.октябрь	2021	01.июль	кв. 2	Текст А	4-й период	Товар 4
5	13:00	пт	Пятница	май	январь	январь.08	15.январь	2022	01.сентябрь	кв. 3	текст 3	5-й период	Товар 5
6	14:00	сб	Суббота	июнь	апрель	апрель.08	15.апрель	2023	01.ноябрь	кв. 4	Текст А	6-й период	Товар 6
7	15:00	вс	Воскресенье	июль	июль	июль.08	15.июль	2024	01.январь		текст 4	7-й период	Товар 7
8	16:00			август	октябрь	октябрь.08	15.октябрь	2025	01.март		Текст А	8-й период	Товар 8
9	17:00			сентябрь	январь						текст 5	9-й период	Товар 9
10	18:00			ноябрь							Текст А	10-й период	Товар 10
				декабрь									

## Задание 2

1. На Листе1 оформите таблицу по образцу. Переименуйте Лист1 в Характеристики авто (рис. 1).
2. Вычислите значение в ячейке D4 ( $=E4+F4$ ). Используя маркер автозаполнения, заполните столбец D.
3. Вычислите значение в ячейке G4 ( $=H4+I4$ ). Используя маркер автозаполнения, заполните столбец G.
4. Используя контекстное меню листа – правой кнопкой по ярлыку - (**переместить/скопировать**), скопируйте лист 1 ещё на 6 листов. Обязательно ставьте галочку в строке **Создавать копию**.
5. **Лист 2** переименуйте в **Статистика**. Вычислите максимальную скорость, минимальный расход топлива, среднее значение снаряженной массы и суммарный расход топлива на 100 км для всех автомобилей (используйте функции **МАКС**, **МИН**, **СРЗНАЧ**, **СУММ**). Для вычисления добавьте еще 4 строки с названиями применяемых функций.
6. На **Листе 3** (Назовите его **Условное форматирование**) выполните условное форматирование полной массы автомобилей (столбец G) исходных данных (вкладка **Главная** → **Условное форматирование** → **правила выделения ячеек** → **условие** → **пользовательский режим**):
  - до 1000 кг – синий цвет текста, начертание – курсив;
  - от 1000 км до 1500 км включительно – залить ячейки желтым цветом;
  - более 1500 км – красный цвет шрифта, начертание – полужирный курсив.
7. На **Листе 4** (**Сортировка**) отсортируйте таблицу в порядке возрастания длины автомобилей (**Данные** → **Сортировка**). Обратите внимание, для того, чтобы не перепутать значения в строках следует выделить все значения, а не только столбец *Длина*.
8. На **Листе 5** (Назовите его **Фильтрация**) используя команду **Данные** → **Фильтр**, определите экономичные автомобили, у которых контрольный расход топлива на 100 км менее 5 литров. Для фильтрации необходимо выделять данные вместе с заголовками, для того, чтобы первая строка не выпадала из анализа.

9. **Лист 6 – Расчеты.** После поля «**Контрольный расход топлива на 100 км**» добавьте поле «**Экономичность**». Если контрольный расход топлива автомобиля на 100 км не более 6 л, присвойте ячейке значение «**экономичный**». Используйте функцию ЕСЛИ. Откройте окно формул, найдите необходимую. Функция состоит из трёх аргументов. В условии указывается анализируемая ячейка и знак сравнения. В значении Истина пишется то, что будет добавлено, если условие выполняется (текст автоматически берется в кавычки), в значении Ложь вариант, если условие не выполняется.

10. На листе **Расчеты**, пользуясь командой **Данные → Проверка**, организовать проверку вводимых значений для поля «**Число посадочных мест**» (действительные значения не меньше 4) с сообщением для ввода и сообщением об ошибке **Предупреждение**.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	<b>Характеристики легковых автомобилей</b>													
2														
3	Марка автомобиля	Ведущий мост	Число посадочных мест	Снаряженная масса	на переднюю ось	на заднюю ось	Полная масса	на переднюю ось	на заднюю ось	Длина	Ширина	Высота	Максимальная скорость, км/ч	Контрольный расход топлива на 100 км, л
4	ВАЗ-1111 "Ока"	передний	4		395	240		495	480	3200	1420	1400	120	4,5
5	ВАЗ-968М "Запорожец"	задний	5		320	520		450	710	3765	1490	1425	118	6
6	ВАЗ-1102 "Таврия"	передний	5		440	270		573	537	3708	1554	1410	140	4,6
7	ВАЗ-2101 "Жигули"	задний	5		515	440		615	740	4073	1611	1440	142	7
8	ВАЗ-2105 "Жигули"	задний	5		545	450		635	760	4130	1620	1446	145	7,1
9	ВАЗ-2121 "Нива"	задний и передний	5		680	470		750	800	3748	1680	1640	132	9,8
10	ВАЗ-2108 "Спутник"	передний	5		560	340		670	655	4006	1620	1402	148	5,7
11	АЗЛК-2140 "Москвич"	задний	5		560	485		670	755	4170	1550	1380	142	7,4
12	АЗЛК-2141 "Москвич"	передний	5		650	420		775	695	4350	1690	1400	155	6,1
13	ИЖ-2126 "Орбита"	задний	5		563	477		677	763	4068	1650	1450	150	6,9
14	УАЗ-31512	задний и передний	7		875	715		1030	1400	4050	1785	2015	110	11,3
15	ГАЗ-2410 "Волга"	задний	5		745	655		760	940	4735	1800	1476	147	9,3
16	ГАЗ-3102 "Волга"	задний	5		760	690		870	980	4960	1820	1476	147	9,3

### 3 Варианты заданий

Варианты заданий отличаются тематикой набираемого текста:

Вариант	Тема
1	Экономика
2	Бухгалтерия
3	Билетная касса
4	Библиотека



5	Фильмотека
6	Погода
7	Автомобили
8	Велосипеды
9	Обучение
10	Животные

## 7. Использование сложных функций табличного процессора. Сводные таблицы, диаграммы, макросы.

### 1 Цель работы

Изучение возможностей табличного процессора. Режимы работы с таблицей. Выполнение статистических и логических функций. Построение диаграмм и сводных таблиц.

### 2 Порядок выполнения работы

#### Задание 1.

Заполнить ячейки недостающими формулами:

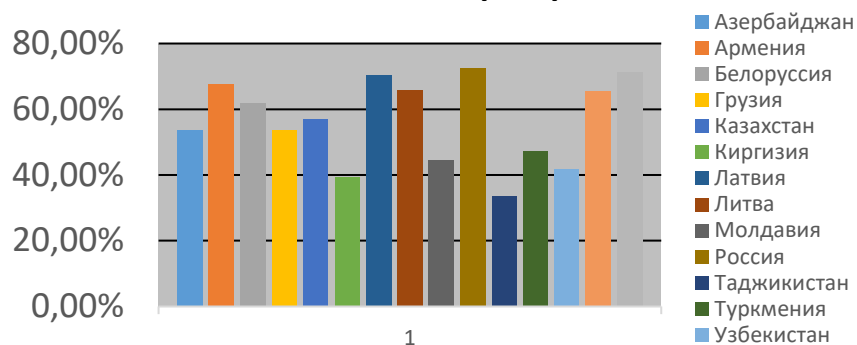
- доля городского населения от населения государства в целом (отношение городского населения к общему);
- плотность населения данного государства (отношение населения к территории);
- суммарные данные по каждому признаку кроме последних двух столбцов (строка «Всего»);
- Создать динамическую таблицу из получившейся;
- Вставить сводную таблицу на новый лист;
- Выстроить необходимые фильтры.

**Примечание.** Все вычисления производить с точностью до десятых.

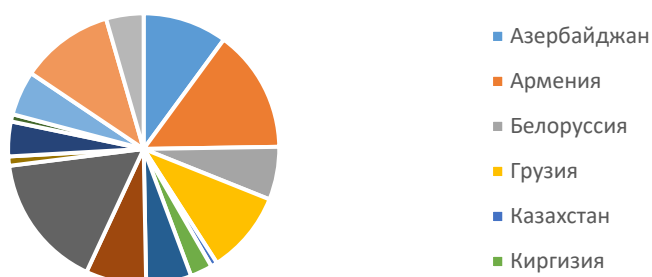
Примеры данных: все вычисления производятся по формулам до десятичных знаков.						
№	Государство	Территория, тыс. км	Население, тыс. чел.	В том числе городское	Доля городского населения, %	Плотность населения, чел. на 1 тыс км <sup>2</sup>
1	Азербайджан	86,6	6614	3553		
2	Армения	29,8	3317	2242		
3	Белоруссия	207,6	9942	6167		
4	Грузия	69,7	5201	2798		
5	Казахстан	2717,3	15842	9045		
6	Киргизия	198,5	3967	1569		
7	Латвия	63,7	2604	1836		
8	Литва	65,2	3570	2345		
9	Молдавия	33,7	4111	1838		
10	Россия	17075,4	143090	103860		
11	Таджикистан	143,1	4499	1512		
12	Туркмения	488,1	3189	1513		
13	Узбекистан	447,4	17974	7530		
14	Украина	603,7	50840	33244		
15	Эстония	45,1	1530	1093		
Всего		=сумм( : )	=сумм( : )	=сумм( : )		
Статистика		=мин( : )	=макс( : )	=срзнач( : )		

## Задание 2. Построить диаграммы:

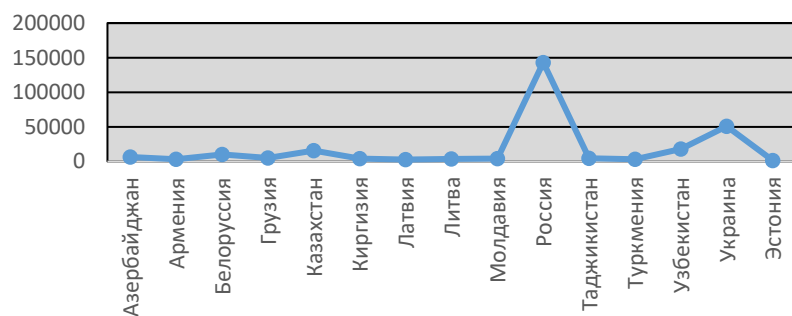
**Доля городского населения в государствах  
постсоветского пространства**



**Плотность населения**

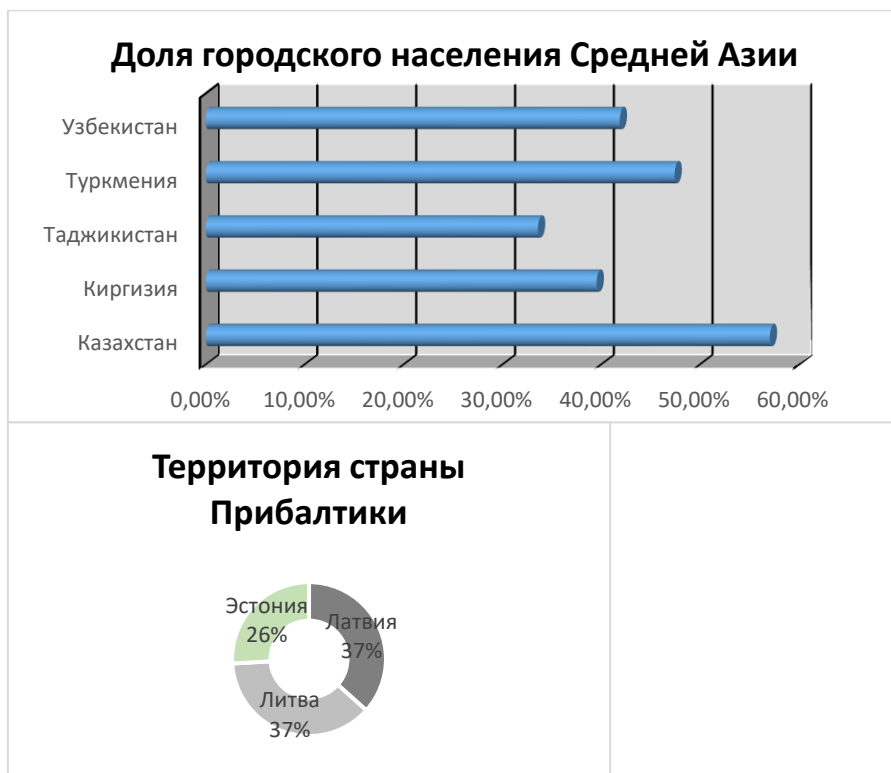


**Население государств постсоветского  
пространства**



**Население России**





### 3 Варианты заданий:

Варианты заданий отличаются тематикой набираемого текста:

Вариант	Тема
1	Экономика
2	Бухгалтерия
3	Билетная касса
4	Библиотека
5	Фильмотека
6	Погода
7	Автомобили
8	Велосипеды
9	Обучение
10	Животные

## **8. Создание мультимедийных презентаций. Создание структуры, формирование слайдов, управление показом презентации.**

### **1 Цель работы**

Изучение основных возможностей программы для создания и проведения презентаций, получение практических навыков по созданию и редактированию презентаций.

### **2 Краткая теория**

Создание презентации предоставляет ряд следующих вариантов для создания новой презентации:

- Создать.

Слайды имеют минимум элементов оформления и цвета к ним не применены.

- Из имеющейся презентации.

Презентация создается на основе уже имеющейся презентации с заданным оформлением. Создается копия имеющейся презентации, позволяющая создать новую презентацию, внося изменения в оформление и содержимое исходной презентации.

- Из шаблона оформления.

Презентация создается на основе имеющегося шаблона, содержащего основные элементы оформления, шрифты и цветовую схему. Кроме стандартных шаблонов можно использовать самостоятельно созданные шаблоны.

- Из мастера автосодержания.

Для применения шаблона оформления, включающего предлагаемый текст для слайдов, используется мастер автосодержимого. Затем в предложенный текст вносятся необходимые изменения.

- Шаблоны на веб-узле.

Создание презентации с помощью шаблона, находящегося на веб-узле.

Можно также вставить слайды из других презентаций или текст из других приложений, например из Libre Writer.

### ***Советы по созданию эффективных презентаций***

Число слайдов должно быть минимальным.

Чтобы четко сформулировать свои мысли и завоевать внимание и интерес аудитории, необходимо включать в презентацию минимальное число слайдов.

Выбирайте такой размер шрифта, чтобы ваши слушатели легко могли прочитать надписи.

Выбор оптимального размера шрифта поможет докладчику донести свои идеи до слушателей. Не забывайте, что аудитория будет читать надписи на слайдах издалека.

В общем случае слушателям может быть трудно разглядеть шрифт с размером менее 18.

Текст слайдов должен быть простым

Вы, вероятно, предпочтете, чтобы люди слушали вас, а не пытались прочитать текст на экране. Используйте маркированные списки или короткие предложения и постарайтесь разместить каждый пункт на одной строке, не перенося текст на следующую строчку.

Некоторые проекторы обрезают края слайдов, поэтому длинные предложения, возможно, будут обрезаны.

Подчеркните свои идеи с помощью графических объектов. Рисунки, диаграммы, графики и рисунки SmartArt создадут зрительные образы, которые помогут слушателям запомнить ваши слова. Используйте выразительное изображение в дополнение к тексту и обращениям на слайдах.

Однако, так же как и с текстом, не размещайте на слайде слишком много графики. Надписи на диаграммах и графиках должны быть понятными

Используйте ровно столько текста, чтобы сделать надписи на диаграмме или графике понятными.

Фон на всех слайдах должен быть неярым и уместным. Выберите привлекательный, уместный шаблон или тему, которые не слишком бросаются в глаза, иначе фон или макет будут отвлекать внимание аудитории от содержания презентации.

Необходимо следить и за тем, чтобы цвета фона и текста были контрастными. Во встроенных темах используется контраст между светлым фоном и темным текстом или, наоборот, темным фоном и светлым текстом.

Проверяйте правописание. Чтобы заслужить уважение своей аудитории и не потерять его, всегда проверяйте правописание в своей презентации.

### **3. Задания**

1. Создайте презентацию на одну из нижеперечисленных тем, состоящую из десяти слайдов. Используйте для слайдов различные макеты:

1. Заголовок и подзаголовок.
2. Заголовок и текст в двух колонках.
3. Заголовок и таблица.

4. Заголовок и две диаграммы различного вида.
  5. Заголовок и два объекта — картинка и рисунок.
  6. Заголовок и организационная диаграмма.
- Добавьте заметки к слайдам.

2. Выберите шаблон оформления для всех слайдов. Для первого слайда назначьте цветовую схему, отличающуюся от одной на прочих слайдах. Измените фон второго слайда.

3. Назначьте различные эффекты анимации для любых трёх слайдов.

4. В режиме сортировщика слайдов продублируйте последние два слайда и вставьте итоговый слайд на основе всех слайдов презентации.

5. Назначьте различные эффекты перехода для всех слайдов.

6. Для первых трёх слайдов установите время автоматической смены — 5 секунд.

7. При просмотре внесите пометки в последний слайд указателем типа «фломастер»

#### **4. Варианты**

1. Дополненная реальность (Microsoft Surface, Google Glass...)
2. Интеллектуальная робототехника (ASIMO, AIBO, Pleo...)
3. Электронные книги (Amazon Kindle, Barnes & Noble Nook, Sony Reader...)
4. Социальные сети (VK, LinkedIn...)
5. Планшеты (Apple iPad, Google Nexus 10...)
6. Смартфоны (Apple iPhone, BlackBerry и Google Android...)
7. Машинное зрение
8. Монетизация сайта (Яндекс.Директ, Google AdSense...)
9. Облачные вычисления (SaaS, PaaS, IaaS...)
10. Распознавание музыки (Shazam, Midomi)
11. Распознавание речи (голосовой поиск Apple Siri, Google Voice...)
12. «Зеленые» технологии
13. Сенсорные экраны (емкостные, резистивные...)
14. Распознавание жестов (мультитач, Microsoft Kinect...)
15. Юзабилити пользовательского интерфейса
16. Коллективный интеллект (краудсорсинг, краудфандинг)
17. Технологии беспроводной передачи данных (Wi-Fi, Li-Fi...)

## 9. Создание и редактирование цифрового изображения. Параметры, слои, инструменты. Использование фильтров и эффектов.

### 1 Цель работы

Научиться работе в графическом редакторе для создания и редактирования цифрового изображения, использования параметров, слоев, инструментов, фильтров и эффектов.

### 2 Задания

#### 1. Цветовая коррекция после сканирования.




Откройте файл предложенный преподавателем.

Выполните коррекцию яркости/контраста с помощью команды **Изображение->Настройка>Яркость/Контраст**.

Выполните цветовую коррекцию с помощью команды **Изображение->Настройка>Цветовой баланс**.

Результат сохраните в виде файла pict2.jpg

#### 2. Слои. Копирование.

1. Откройте документ, предложенный преподавателем для фоновой картинки коллажа.
2. Скопируйте все (**Выделить>Все**) содержимое командой **Правка>Копировать**.
3. Создайте новый документ с цветовым режимом RGB и прозрачным фоном.
4. ставьте изображение командой **Правка>Вставить**.
5. Откройте следующий файл. Выделите часть изображения с помощью инструментов выделения (, , ).
6. Измените размер и угол поворота вставленного фрагмента с помощью команды **Правка>Свободное преобразование**.
7. Используйте различные фильтры и эффекты (не менее 4-х), желательно применить к разным слоям.
8. Поработайте с наложением слоев. Сделайте красивый коллаж.
9. Склейте все слои командой **Слой>Простое изображение**.
10. Сохраните изображение в файле pict3.jpg.

#### 3. Слои. Рисование

- 1 Создайте новый документ с размерами 10x15 см, разрешением 72 dpi, цветовым режимом RGB и прозрачным фоном
- 2 Создайте новый слой (см. ниже) **Слой > Новый > Слой**








- 3 На этом слое нарисуйте произвольный рисунок зеленого цвета с помощью инструмента Кисть.
- 4 Используйте различные фильтры и эффекты (не менее 4-х), желательно применить к разным слоям.

Выбор цвета:

двойной щелчок на значке цвета на палитре инструментов и выбрать цвет

либо: палитра **Каталог** - выбрать цвет

либо: палитра **Цвета** - подобрать цвет

- 5 Измените цвет (см. выше) и толщину кисти (Палитра **Кисти** - выбрать тип кисти) и нарисуйте произвольный рисунок
- 6 Выделите область любым инструментом выделения и залейте желтым цветом, предварительно выбрав его на палитре **Каталог**, с помощью инструмента **Заливка** . Поэкспериментируйте со значениями прозрачности заливки в палитре **Опции**.
- 7 Перейдите в слой Фон (Палитра **Слои**), выберите  инструмент **Линейный Градиент**. В палитре **Опции** выбираем тип градиента **От основного цвета к фоновому**.
- 8 На изображении показываем мышью начало и конец вектора градиента. Если вы хотите использовать цвет, уже имеющийся на рисунке, выберите инструмент  Пипетка и щелкните мышью на выбранном цвете. Основной цвет изменится на выбранный.

### 3 Варианты

В качестве вариантов заданий, преподавателем предоставляются исходные графические файлы, в индивидуальном порядке.

## **10. Создание аудио и видео файлов. Нелинейный монтаж, наложение эффектов перекодирование формата.**

### **1 Цель работы**

Ознакомление с процессами создания и редактирования медиа файлов. Изучение основ нелинейного монтажа.

### **2 Задание**

Используя готовые видео и аудио файлы произвести их редактирование при помощи видео и аудио редактора по выбору преподавателя:

- 1 Произвести сжатие файлов в другой формат
- 2 Произвести обрезку файлов по времени
- 3 Используйте разные параметры сжатия и конвертирования медиа файлов. (1 исходник, 2-3 файла выходных)
- 4 Произвести наложение звука на видео
- 5 Произвести монтаж из разных роликов и разных отрезков видео файла.
- 6 Произвести наложение текста (заголовков, титры, комментарии) и эффектов перехода между вставками на видео.

### **3 Варианты**

В качестве вариантов заданий, преподавателем предоставляются исходные видео файлы, в индивидуальном порядке.

## **11. Антивирусная защита. Использование антивирусного ПО.**

### **1 Цель работы**

Ознакомление со спецификой работы антивирусного программного обеспечения. Получение навыка работы с загрузочными дисками (загрузочными антивирусами). Поиск вирусов, анализ антивирусной статистики.

### **2 Краткая теория**

Различают такие типы антивирусных программ:

- 1) **программы-детекторы:** предназначены для нахождения зараженных файлов одним из известных вирусов. Некоторые программы-детекторы могут также лечить файлы от вирусов или уничтожать зараженные файлы. Существуют специализированные, то есть предназначенные для борьбы с

одним вирусом детекторы и полифаги, которые могут бороться с многими вирусами;

2) **программы-лекари**: предназначены для лечения зараженных дисков и программ. Лечение программы состоит в изъятии из зараженной программы тела вируса. Также могут быть как полифагами, так и специализированными;

3) **программы-ревизоры**: предназначены для выявления заражения вирусом файлов, а также нахождение поврежденных файлов. Эти программы запоминают данные о состоянии программы и системных областей дисков в нормальном состоянии (до заражения) и сравнивают эти данные в процессе работы компьютера. В случае несоответствия данных выводится сообщение о возможности заражения;

4) **лекари-ревизоры**: предназначены для выявления изменений в файлах и системных областях дисков и, в случае изменений, возвращают их в начальное состояние.

5) **программы-фильтры**: предназначены для перехвата обращений к операционной системе, которые используются вирусами для размножения и сообщают об этом пользователя. Пользователь может разрешить или запретить выполнение соответствующей операции. Такие программы являются резидентными, то есть они находятся в оперативной памяти компьютера.

6) **программы-вакцины**: используются для обработки файлов и boot-секторов с целью предупреждения заражения известными вирусами (в последнее время этот метод используется все чаще).

### 3 Порядок выполнения работы

С помощью антивирусной программы, записанной на загрузочный диск или флеш накопитель, загрузиться в стороннюю операционную систему (Kaspersky Rescue Disk), обновить вирусные базы, проверить компьютер на наличие вирусов и при их обнаружении вылечить зараженные файлы, собрать статистику проверки.

Чтобы запустить проверку компьютера и удалить вредоносные программы с вашего компьютера, выполните следующие действия:

1. Скачайте образ загрузочного диска с антивирусной утилитой, например Kaspersky Rescue Disk и запишите его на загрузочный диск или USB флеш накопитель.
2. Загрузите компьютер с Kaspersky Rescue Disk 10 (бесплатная версия ПО) в графическом режиме.
3. В левом нижнем углу экрана нажмите на кнопку в виде буквы К. В меню выберите Kaspersky Rescue Disk.
4. Обновите антивирусные базы Kaspersky Rescue Disk. Для этого на закладке Обновление нажмите на кнопку *Выполнить обновление*.
5. Выполните проверку всех локальных дисков на наличие вирусов

## 6. Запишите статистику проверки.



4. Дождитесь завершения обновления антивирусных баз программы.

5. На закладке Проверка объектов установите флажки напротив объектов, которые должна проверить программа. По умолчанию Kaspersky Rescue Disk проверяет загрузочные секторы жестких дисков, а также скрытые объекты автозапуска операционной системы.

6. Нажмите на кнопку *Выполнить проверку объектов*.



7. После окончания проверки в случае обнаружения угроз программа запросит вас, какое действие произвести над вредоносными объектами:

1. *Лечить*. После лечения с объектом можно продолжить работу.

2. *Поместить на Карантин*, если в результате проверки не удалось определить, заражен объект или нет. Если у вас установлена необходимая опция проверки файлов на карантине после каждого обновления баз, то после получения новой сигнатуры лечения объект на Карантине будет вылечен и вновь доступен для пользователя.
3. *Удалить*. Если объекту присвоен статус вируса, но его лечение невозможно, вы можете удалить его. Информация об объекте сохранится в отчете об обнаруженных угрозах.

## 12. Практическое применение криптографии. Шифр Вернама. Помехоустойчивое кодирование. Код Хэмминга.

### 1 Цель работы

Ознакомление с управлением процессами маршрутизации пакетов, передаваемых через сеть. Изучение теории выбора кратчайших путей и ее методов.

### 2 Краткая теория

#### 2.1 Шифр Вернама

**Шифр Вернама** был предложен в 1917 г. телеграфистом Гильбертом Вернамом. Суть **шифра Вернама** том, что зашифрованный текст есть объединение открытого текста и ключа с помощью исключающего ИЛИ (то есть  $\text{EncryptedText} = \text{OpenText} \text{ XOR } \text{Key}$ ). Ключ - случайная последовательность бит. Вернам построил телеграфный аппарат, который выполнял эту операцию после подачи ленты с ключом.

Вернам заметил, что для обеспечения стойкости шифра каждая лента должна быть одноразовой. Это непросто применить на практике, потому возможно закольцовывать несколько лент, причём их длины различны и являются взаимно простыми. Например,

0	1	0	1	1	0	1	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---

Как видим, в зелёной и жёлтой лентах нет закономерностей в размещении бит, при этом длины 7 и 5 - взаимно простые. Когда эта единая большая лента заканчивается, вновь начинаем с левого бита зелёной ленты. При практическом применении шифра Вернама, разумеется, лучше больше лент и длины побольше.

Возможно также применение так называемой схемы шифроблокнотов. Стороны, которые обмениваются сообщениями, имеют в своём

распоряжении по одинаковому блокноту. На каждой страничке шифроблокнота - по ключу. После отправки очередного сообщения обе стороны вырывают страничку из своего блокнота, на которой был записан текущий ключ. Для следующего сообщения будет использоваться ключ со следующей страницы, которая потом так же будет вырвана. В современных условиях в качестве шифроблокнотов могут быть файлы на носителях электронной информации - тогда каждый раз удаляется очередной использованный ключ: все ключи одноразовые.

Проблема в том, что шифроблокноты нужно как-то передать. Для этого нужна личная встреча или надёжный посредник. Чего нельзя точно - так это передавать шифроблокнот по тому же каналу, что будет использоваться для передачи сообщений. На крайняк следует использовать два канала - один для ключей, другой для сообщений. Тогда противнику придётся прослушивать два канала, чтобы раскрыть передаваемые сведения, что несколько усложняет его задачу.

## 2.2 Помехоустойчивое кодирование. Код Хэмминга.

Код Хэмминга состоит из двух частей. Первая часть кодирует исходное сообщение, вставляя в него в определённых местах контрольные биты (вычисленные особым образом). Вторая часть получает входящее сообщение и заново вычисляет контрольные биты (по тому же алгоритму, что и первая часть). Если все вновь вычисленные контрольные биты совпадают с полученными, то сообщение получено без ошибок. В противном случае, выводится сообщение об ошибке и при возможности ошибка исправляется.

В данном примере описан самый простой алгоритм Хемминга, который может исправлять лишь одну ошибку (существуют более совершенные модификации данного алгоритма, которые позволяют обнаруживать (и если возможно исправлять) большее количество ошибок).

### Подготовка сообщения

Допустим, у нас есть сообщение «habr», которое необходимо передать без ошибок. Для этого сначала нужно наше сообщение закодировать при помощи Кода Хэмминга. Нам необходимо представить его в бинарном виде.

Символ	ASCII код	Бинарное представление
h	68	01000100
a	61	00111101
b	62	00111110
r	72	01001000

На этом этапе стоит определиться с, так называемой, длиной информационного слова, то есть длиной строки из нулей и единиц, которые мы будем кодировать. Допустим, у нас длина слова будет равна 16. Таким образом, нам необходимо разделить наше исходное сообщение («habr») на блоки по 16 бит, которые мы будем потом кодировать отдельно друг от друга. Так как один символ занимает в памяти 8 бит, то в одно кодируемое слово помещается ровно два ASCII символа. Итак, мы получили две бинарные строки по 16 бит:

h	a	и	b	r
01000100	00111101		00111110	01001000

После этого процесс кодирования распараллеливается, и две части сообщения («ha» и «br») кодируются независимо друг от друга. Рассмотрим, как это делается на примере первой части.

Прежде всего, необходимо вставить контрольные биты. Они вставляются в строго определённых местах — это позиции с номерами, равными степеням двойки. В нашем случае (при длине информационного слова в 16 бит) это будут позиции 1 ( $2^0$ ), 2 ( $2^1$ ), 4 ( $2^2$ ), 8 ( $2^3$ ), 16 ( $2^4$ ). Соответственно, у нас получилось 5 контрольных бит (подчеркнуты):

<b>Было:</b>	h	a	<b>Стало:</b>	h	a
	01000100	00111101		00001000	001011101

Таким образом, длина всего сообщения увеличилась на 5 бит. До вычисления самих контрольных бит, мы присвоили им значение «0».

### Вычисление контрольных бит

Теперь необходимо вычислить значение каждого контрольного бита. Значение каждого контрольного бита зависит от значений информационных бит, но не от всех, а только от тех, которые этот контрольный бит контролирует. Для того чтобы понять, за какие биты отвечает каждый контрольный бит необходимо понять очень простую закономерность: контрольный бит с номером N контролирует все последующие N бит через каждые N бит, начиная с позиции N:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1	
x		x		x		x		x		x		x		x		x		x		x	1
	x	x			x	x			x	x			x	x			x	x			2
			x	x	x	x					x	x	x	x					x	x	4
							x	x	x	x	x	x	x	x							8
															x	x	x	x	x	x	16

Здесь знаком «X» обозначены те биты, которые контролирует контрольный бит, номер которого справа. То есть, к примеру, бит номер 12 контролируется битами с номерами 4 и 8. Ясно, что чтобы узнать какими битами контролируется бит с номером N надо просто разложить N по степеням двойки.

Дальше вычисляем значение каждого контрольного бита: берём каждый контрольный бит и смотрим сколько среди контролируемых им битов единиц, получаем некоторое целое число и, если оно чётное, то ставим ноль, в противном случае ставим единицу. Можно конечно и наоборот, если число чётное, то ставим единицу, в противном случае, ставим 0. Главное, чтобы в «кодирующей» и «декодирующей» частях алгоритм был одинаков. (Мы будем применять первый вариант). Высчитав контрольные биты для нашего информационного слова получаем следующее:

h	a
10011000	0100 001011101

и для второй части:

b	r
10010110	1110 010101000

### Декодирование и исправление ошибок

Теперь, допустим, мы получили закодированное первой частью алгоритма сообщение, но оно пришло к нам с ошибкой. К примеру, мы получили такое (11-ый бит передался неправильно):

h	a
1001100001	10 001011101

Вся вторая часть алгоритма заключается в том, что необходимо заново вычислить все контрольные биты (так же как и в первой части) и сравнить их с контрольными битами, которые мы получили. Так, посчитав контрольные биты с неправильным 11-ым битом мы получим такую картину:

h	a
0101100101	10 001011101

В данном случае, контрольные биты под номерами: 1, 2, 8 не совпадают с такими же контрольными битами, которые мы получили. Теперь просто сложив номера позиций неправильных контрольных бит ( $1 + 2 + 8 = 11$ ) мы получаем позицию ошибочного бита. Теперь просто инвертировав его и отбросив контрольные биты, мы получим исходное сообщение в первоизданном виде! Абсолютно аналогично поступаем со второй частью сообщения.



### Примечания

1. В данном примере используется длина информационного сообщения 16 бит, но длину можно взять любую.
2. Для каждого числа проверочных символов используется специальная маркировка вида  $(k,i)$ , где  $k$  — количество символов в сообщении,  $i$  — количество информационных символов в сообщении. Например, существуют коды  $(7, 4)$ ,  $(15, 11)$ ,  $(31, 26)$ .
3. Число контрольных символов ( $r$ ) можно вычислить по эмпирической формуле:

$$r = [\log\{(i+1) + [\log(i+1)]\}], \text{ где}$$

$i$  - количество информационных символов в сообщении;

$[\cdot]$  - означает округление до большего ближайшего целого значения

4. Каждый проверочный символ в коде Хэмминга представляет сумму по модулю 2 некоторой подпоследовательности данных. Стоит учитывать, что в данной простой версии алгоритма на одно информационное слово можно исправить только одну ошибку.
5. Код  $(7,4)$  является минимально возможным кодом с достаточно большой **избыточностью**.

Эффективность кода  $(k/n)$  растет с увеличением длины кода

6. **Избыточность кода** — это количество проверочной информации в сообщении. Рассчитывается она по формуле:

$k/(i+k)$ , где

$k$  — количество проверочных бит,

$i$  — количество информационных бит.

Например, мы передаем 3 бита и к ним добавляем 1 проверочный бит — избыточность составит  $1/(3+1) = 1/4$  (25%).

### 4 Задание

1. Разработать программу для шифрования и расшифрования простых текстовых сообщений по методу Вернама. Зашифровать дату своего рождения ключом из таблицы, согласно своему варианту.

Входные данные:

Вариант	1	2	3	4	5	6	7	8	9	10
Ключ	1831	2403	2494	2642	2724	3257	2251	4179	1253	2935

2. Зашифрованную дату рождения представить в двоичном виде и закодировать с помощью помехоустойчивого кода Хемминга. Длина кодового слова не менее 16 бит.
3. Реализовать проверку, обнаружение и исправление одиночной ошибки в заново введённом кодированном сообщении по методу Хэмминга.