Некоммерческое акционерное общество

АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра компьютерной и инфокоммуникационной безопасности

ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ

Методические указания по выполнению расчетно-графических работ
для студентов специальности
5В100200 – Системы информационной безопасности

Алматы 2015

СОСТАВИТЕЛЬ: Е.А. Зуева. Программирование на языках высокого уровня. Методические указания к выполнению расчетно-графических работ для студентов специальности 5В100200 - Системы информационной безопасности. - Алматы: АУЭС, 2015. – 30 с.

Методические указания содержат указания по подготовке и выполнению расчетно-графических работ, целью которых является изучение основ программирования на языке высокого уровня Python; приведены описания расчетно-графических работ, дана методика проведения и ход выполнения, оговорен перечень рекомендуемой литературы.

Методические указания предназначены для студентов всех форм обучения специальности 5В100200 – Системы информационной безопасности.

Илл. 7, табл. 20, библиогр. - 6 назв.

Рецензент: доцент Ползик Е.В.

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2015г.

Содержание

Введение

В настоящий сборник включены расчетно-графические работы, целью которых является приобретение навыков программирования и реализации различных алгоритмов действий на компьютере, в том числе получение практических навыков работы с таким высокоуровневым языком программирования, как Python.

Дисциплина «Программирование на языках высокого уровня» является дисциплиной по выбору, и самостоятельное выполнение расчетно-графических работ (РГР) по данному курсу дает возможность отработки навыков и наработки знаний у студентов. Для закрепления материала в курсе предлагается выполнить 3 расчетно-графические работы.

Все расчетно-графические работы ориентированы на проявление элементов научно-исследовательской деятельности студентов.

Выполнение каждой расчетно-графической работы должно завершаться оформлением отчета, согласно [1]. Выполненная работа и оформленный отчет защищается у преподавателя.

Приведены описания каждой работы, дана методика проведения, оговорен перечень рекомендуемой литературы.

В рабочем задании указаны варианты тем выполнения для каждого студента. Номер варианта выбирается согласно порядковому номеру в общем списке группы журнала преподавателя.

1 Расчетно-графическая работа №1. Простые методы шифрования

Цель работы: изучение реализации алгоритмов шифрования класса простых шифров.

1.1 Рабочее задание

Зашифровать следующий текст: «Криптография - тайнопись, система изменения письма с целью сделать текст понятным для лиц, знающих эту систему. Стеганография – наука о скрытой передаче информации путем сохранения в тайне самого факта.», имея индивидуальные ключи с помощью методов:

1 - шифра Цезаря;
2 - шифра Цезаря с ключевым словом;
3 - афинной криптосистемы;
4 - доски Полибия;
5 - шифра Вижинера;
6 - шифра Гронсфельда;
7 - шифра биграммами;
8 - шифра биграммами с двойным квадратом;
9 - шифра квадратом Кардано;
10 - перестановочным шифром с ключевым словом.

При этом ключом является число (порядковый номер студента в списке группе), и если необходимы буквы, то надо взять фамилия и имя студента.

Если метод не предусматривает шифрование целой фразы, то допускается сокращение фразы до «Криптография – тайнопись, стеганография – сокрытие факта передачи». В большинстве методов регистр букв не учитывать и потому выставлять так же, как и в исходной фразе.

1.2 Методические указания к выполнению работы

Общее описание методов: методы 1-4 – методы одноалфавитной замены; методы 5-6 – методы многоалфавитной замены; методы 7-8 – биграммные (шифруется по 2 буквы за раз); методы 9-10 – перестановочные. Недостатки каждой из групп:

а) методы 1-4: шифр Цезаря и аффинная криптосистема относятся к классу одноалфавитных криптосистем, то есть для выбранного ключа некоторая буква исходного открытого текста всегда будет заменяться одной и той же буквой в шифротексте. Поэтому данные шифры могут быть вскрыты методом частотного криптоанализа. Частотный анализ использует то свойство зашифрованного текста, что частота встречаемости символов в нем совпадает с частотой встречаемости соответствующих символов в открытом тексте. Если же учесть, что частоты встречаемости различных символов в текстах соответствующего языка распределены неравномерно (так, например, относительная частота встречаемости буквы «А» в текстах на русском языке составляет 0.069, а буквы «Ф» 0.003), то, подсчитав относительную частоту

встречаемости букв в шифротексте, можно предположить, что символ, наиболее часто встречающийся в шифротексте, соответствует символу, наиболее часто встречающемуся в текстах на соответствующем языке, и найти таким образом ключ k для шифра Цезаря. Для вскрытия параметров a и b аффинной криптосистемы потребуется найти соответствие двух букв – наиболее часто встречающейся в шифротексте и второй по частоте. Эффективность частотного анализа шифра Цезаря с ключевым словом во многом зависит от длины используемой ключевой фразы;

б) методы 5-6: в шифре Вижинера одной и той же букве шифротекста могут соответствовать различные символы открытого текста в зависимости от того, каким символом ключа они были замаскированы. Это делает бессмысленным подсчет частоты встречаемости символов в шифротексте. Для криптоанализа можно использовать метод Казиски: определять длину парольной фразы по расстоянию между одинаковыми фрагментами шифротекста. Допустим, найдены 2 одинаковых фрагмента шифротекста, расстояние между которыми составляет 20 символов. Это может означать, что 2 одинаковых фрагмента открытого текста были зашифрованы с одной и той же позиции ключа. Это позволяет предположить, что парольная фраза имеет длину 4, 5, 10 или 20 символов. Узнав (или угадав) длину парольной фразы m, можно осуществить частотный криптоанализ шифротекста для выборки каждого m-го символа шифротекста. Шифр Гронсфельда повторяет процедуру шифрования Вижинера, но вместо порядкового номера символа ключа в алфавите использует непосредственное десятичное значение (недостатком является уменьшение величины сдвига для каждого символа величиной 9);

в) методы 7-8: криптостойкость биграммных методов существенно выше, чем у методов простой замены (метод биграмм с двойным квадратом использовался военными еще во времена Второй мировой войны). Частотный анализ здесь можно применить только для оценки частоты встречаемости тех или иных буквосочетаний. И хотя анализ показывает, что какие-то биграммы встречаются в текстах на заданном языке чаще других, большое количество биграмм (а их насчитывается 1089 только для букв русского языка) не позволяет выявить соответствие биграмм открытого и закрытого текста. Криптоаналитику остается только перебирать все возможные варианты расположения символов в таблице;

г) методы 9-10: для дешифрации сообщения необходимо иметь точную копию квадрата, использовавшегося при шифровании (расположение прорезей на квадрате и составляет ключ). Количество разных вариантов расположения прорезей в квадрате nxn равно $4^{\frac{N^2}{4}}$, что для квадрата 6x6 дает 262144 варианта (эквивалент 18-битного ключа). Однако, данный шифр (как впрочем, все перестановочные) ослабляет то, что при криптоанализе можно использовать особенности фонетики национального языка (наиболее часто встречающиеся или недопустимые для данного языка комбинации символов, средняя длина слов и т.п.).

Ниже для шифрования берется текст: «Криптография - тайнопись, специальная система изменения обычного письма, используемая с целью сделать текст понятным лишь для ограниченного числа лиц, знающих эту систему. Применялась для зашифровки военных, дипломатических, торгово-финансовых, нелегально-политических, религиозно-еретических текстов.».

1. Шифр Цезаря.

Шифр Цезаря - один из наиболее древнейших известных шифров. Каждой букве алфавита сопоставляется число, как показано в таблице 1.

Таблица 1 - Алфавитное представление букв

1 а	2 б	3 в	4 г	5 д	6 е	7 ё	8 ж	9 з	10 и	11 й
12 к	13 л	14 м	15 н	16 о	17 п	18 р	19 с	20 т	21 у	22 ф
23 х	24 ц	25 ч	26 ш	27 щ	28 ъ	29 ы	30 ь	31 э	32 ю	33 я

Схема шифрования: берется каждая буква исходной фразы и используется сдвиг буквы алфавита на фиксированное число позиций (k) вперед(+) или назад(-) по алфавиту, буквы в конце алфавита преобразуются в буквы начала алфавита. Неалфавитные символы - знаки препинания, пробелы, цифры - не меняются. Замена букв со сдвигом +1 проиллюстрирована на рисунке 1.

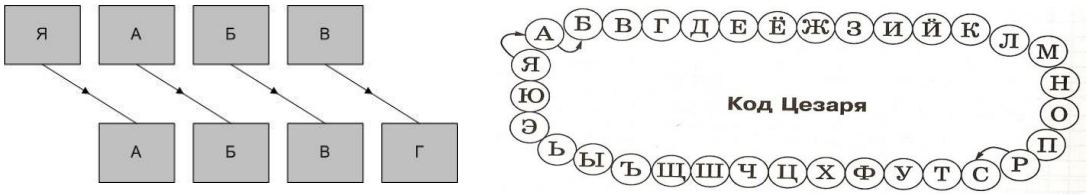


Рисунок 1 – Замена букв со сдвигом +1.

Пояснение на примере. Будем шифровать фразу «сокрытие информации». При k=+1 шифрованное примет вид: «тплсьуйё йохпснбчйй»; при k=-1 фраза будет «рнйпъсзд змунпляхзз»; при k=-3 фраза выглядит «олзншпёв ёкслнйэуёё»; при k=+2 зашифрованное будет выглядеть: «урмтэфкж кпцртовшкк».

Шифр является крайне слабым, и исходный текст можно восстановить, проверив все возможные преобразования и совпадения по словарю маленькой части теста.

Пример расчета. Если, например, вариант студента = 9, знак «+» или «-» студент выбирает сам, то в итоге при смещении +9 исходная фраза: «Криптография - тайнопись, специальная система изменения обычного письма, используемая с целью сделать текст понятным лишь для ограниченного числа лиц, знающих эту систему. Применялась для зашифровки военных, дипломатических, торгово-финансовых, нелегально-политических, религиозно-еретических текстов.».

Зашифрованная фраза: «Ущсшычлщиэсз – ыитцчшсъе, ъшнясифециз ъсъынхи срхнцнцсз чйдацчлч шсъехи, съшчферьнхиз ъ янфеж ъмнфиые ынуъы шчцзыцдх фсбе мфз члщицсанццчлч асъфи фся, рцижвсю ёыь ъсъынхь. Шщсхнцзфчъе мфз рибсэщчкус кчнццдю, мсшфчхиысанъусю, ычщлчкч-эсцицъчкдю, цнфнлифецч-шчфсысанъусю, щнфслсчрцч-нщнысанъусю ынуъычк.».

2. Шифр Цезаря с ключевым словом.

В данной разновидности шифра Цезаря используются число k и ключ шифрования. Создаем таблицу замен. Пояснение на примере k=3, фраза «Фамилиястудента». Выписываем алфавит, а под ним, начиная с k-й позиции, ключ шифрования без повтора букв (таблица 2), оставшиеся буквы записываются в алфавитном порядке после ключевого слова – таблица 3; в итоге - таблица 4. В итоге мы получаем подстановку для каждой буквы. При шифровании: «сокрытие информации» получается «звджщйти тбовжнюртт».

Таблица 2 - Вписывание фразы в исходный алфавит

1 а	2 б	3 в ф	4 г а	5 д м	6 е и	7 ё л	8 ж я	9 з с	10 и т	11 й у	12 к д	13 л е	14 м н	15 н	16 о	17 п
18 р	19 с	20 т	21 у	22 ф	23 х	24 ц	25 ч	26 ш	27 щ	28 ъ	29 ы	30 ь	31 э	32 ю	33 я	

Таблица 3 - Заполнение таблицы оставшимися символами алфавита

1 а	2 б	3 в ф	4 г а	5 д м	6 е и	7 ё л	8 ж я	9 з с	10 и т	11 й у	12 к д	13 л е	14 м н	15 н б	16 о в	17 п г
18 р ж	19 с з	20 т й	21 у к	22 ф о	23 х п	24 ц р	25 ч х	26 ш ц	27 щ ч	28 ъ ш	29 ы щ	30 ь ъ	31 э ы	32 ю ь	33 я э	

Таблица 4 - Создание конечного алфавита

1 а ю	2 б я	3 в ф	4 г а	5 д м	6 е и	7 ё л	8 ж я	9 з с	10 и т	11 й у	12 к д	13 л е	14 м н	15 н б	16 о в	17 п г
18 р ж	19 с з	20 т й	21 у к	22 ф о	23 х п	24 ц р	25 ч х	26 ш ц	27 щ ч	28 ъ ш	29 ы щ	30 ь ъ	31 э ы	32 ю ь	33 я э	

Пример расчета. Фамилия и имя студентки – Толуспаева Данагуль, порядковый номер по списку группы – 9, ключ записывается без повторения букв: «Толуспаевднгь», k=9. Из таблицы первоначального алфавита

1 а	2 б	3 в	4 г	5 д	6 е	7 ё	8 ж	9 з	10 и	11 й
12 к	13 л	14 м	15 н	16 о	17 п	18 р	19 с	20 т	21 у	22 ф
23 х	24 ц	25 ч	26 ш	27 щ	28 ъ	29 ы	30 ь	31 э	32 ю	33 я

получаю таблицу конечного алфавита, после сдвига и преобразования с ключом:

1 ч	2 ш	3 щ	4 ъ	5 ы	6 э	7 ю	8 я	9 т	10 о	11 л
12 у	13 с	14 п	15 а	16 е	17 в	18 д	19 н	20 г	21 ь	22 б
23 ё	24 ж	25 з	26 и	27 й	28 к	29 м	30 р	31 ф	32 х	33 ц

Исходная фраза: «Криптография - тайнопись, специальная система изменения обычного письма, используемая с целью сделать текст понятным лишь для ограниченного числа лиц, знающих эту систему. Применялась для зашифровки военных, дипломатических, торгово-финансовых, нелегально-политических, религиозно-еретических текстов.».

Зашифрованная фраза: «Удовгеъдчбоц – гчлаевонр, нвэжочсрачц нонгэпч отпэаэаоц ешмзаеъе вонрпч, онвесртьэпчц н жэсрх ныэсчгр гэунг веацгамп соир ысц еъдчаозэааеъе зонси сож, тачхйоё фгь нонгэпь. Вдопэацсчнр ысц тчиобдещуо щеэаамё, ыовсепгозэнуоё, гедъеще боачанещмё, аэсэчсрае-весогозэнуоё, дэсоъоетае-эдэгозэнуоё гэунгещ.».

3. Афинная криптосистема.

Обобщением системы Цезаря является аффинная криптосистема. Она определяется двумя числами a и b, где 0<=a, b<=n-1, n - мощность алфавита (количество символов в нем). Числа *a* и *n* должны быть взаимно просты. Соответствующими заменами являются: Aa,b(j)=(aj+b)(mod n) и A-1a,b(j)=(j-b+n)*a-1(mod n). Поиск мультипликативно-обратного (обозначенного как a-1) осуществляется по алгоритму Евклида. Очевидно, что при a=1 аффинная криптосистема вырождается в шифр Цезаря.

Создаем таблицу замен. Пояснение на примере (таблица 5): мощность алфавита – 33 буквы, выбираем произвольно взаимно простые числа 2 и 5. Тогда формула (2·i+5)mod33, где i-порядковый номер буквы в алфавите (например, а=0, б=1 и т. д.): 2·0+5=5(Е); 2·1+5=7(Ж) и т.д. (таблица 6). Таким образом, при шифровании: «сокрытие информации» получается «звъёыйцо цанвёюесцц».

Таблица 5 - Начальная таблица символов алфавита

0 а	1 б	2 в	3 г	4 д	5 е	6 ё	7 ж	8 з	9 и	10 й	11 к	12 л	13 м	14 н	15 о	16 п
17 р	18 с	19 т	20 у	21 ф	22 х	23 ц	24 ч	25 ш	26 щ	27 ъ	28 ы	29 ь	30 э	31 ю	32 я	

Таблица 6 - Рассчитанная таблица значений букв

5 е	7 ж	9 и	11 к	13 м	15 о	17 р	19 т	21 ф	23 ц	25 ш	27 ъ	29 ь	31 ю	0 а	2 в	4 д
6 ё	8 з	10 й	12 л	14 н	16 п	18 с	20 у	22 х	24 ч	26 щ	28 ы	30 э	32 я	1 б	3 г	

Пример расчета. Вариант студента – 5, были взяты значения: a=2, b=5. Преобразование E=a·t+b, НОД (33, 2)=1, где t-порядковый номер буквы в алфавите (например, а=0, б=1 и т. д.): 2·0+5=5(Е); 2·1+5=7(Ж); 2·2+5=9(И). Первоначальная таблица:

0 а	1 б	2 в	3 г	4 д	5 е	6 ё	7 ж	8 з	9 и	10 й
11 к	12 л	13 м	14 н	15 о	16 п	17 р	18 с	19 т	20 у	21 ф
22 х	23 ц	24 ч	25 ш	26 щ	27 ъ	28 ы	29 ь	30 э	31 ю	32 я

Полученная таблица:

5 е	7 ж	9 и	11 к	13 м	15 о	17 р	19 т	21 ф	23 ц	25 ш
27 ъ	29 ь	31 ю	0 а	2 в	4 д	6 ё	8 з	10 й	12 л	14 н
16 п	18 с	20 у	22 х	24 ч	26 щ	28 ы	30 э	32 я	1 б	3 г

Исходная фраза: «Криптография - тайнопись, специальная система изменения обычного письма, используемая с целью сделать текст понятным лишь для ограниченного числа лиц, знающих эту систему. Применялась для зашифровки военных, дипломатических, торгово-финансовых, нелегально-политических, религиозно-еретических текстов.».

Зашифрованная фраза: «Ъёцдйвкёенцг - йешавдцзщ здо,сцеьэаег зцзйоюе цфюоаоацг вжыуавкв дцзэюе, цздвьэфлоюег з соьэб змоьейэ йоъзй двагаыю ьцхэ мьг вкёеацуоаавкв уцзье ьцс, фабчцп яйл зцзйоюл. Дёцюоагьезэ мьг фехцнёвиъц ивоааып, мцдьвюейцуозъцп, йвёквив-нцаеазвып, аоьокеьав-двьцйцуозъцп, ёоьцкцвфав-оёойцуозъцп йоъзйви.».

4. Доска Полибия.

Рассмотрим прямоугольник, размер которого 12·3 (таблица 7). Каждую букву фразы для шифрования заменим на 2 буквы – указание строки и столбца, на пересечении которых стоит рассматриваемая буква, например, «а» заменяется на «АА», «д» на «АД», «с» на «БЁ», «пробел» на «ВИ». При шифровании: «сокрытие информации» получается «БЁБГАКБЕВДБЖАИАЕ ВИАИБВБИБГБЕББААБКАИАИ». При меньших размерах досок допускается сокращение количества букв (то есть убираем буквы ё, й, ь, ъ, например, но пробел должен присутствовать так как он является разделителем слов).

Пример расчета. Вариант – 9, размер доски Полибия был выбран 9·4, таблицы 7, 8.

Таблица 7 - Доска Полибия 12·3

	А	**Б**	**В**	**Г**	**Д**	**Е**	**Ё**	**Ж**	**З**	**И**	**Й**	**К**
А	а	б	в	г	д	е	ё	ж	з	и	й	к
Б	л	м	н	о	п	р	с	т	у	ф	х	ц
В	ч	ш	щ	ъ	ы	ь	э	ю	я	пробел	.	,

Таблица 8 - Доска Полибия 9*4

	А	Б	В	Г	Д	Е	Ж	З	И
А	а	б	в	г	д	е	ж	з	и
Б	й	к	л	м	н	о	п	р	с
В	т	у	ф	х	ц	ч	ш	щ	ъ
Г	ы	ь	э	ю	я		,	-	.

Исходная фраза: «Криптография - тайнопись, специальная система изменения обычного письма, используемая с целью сделать текст понятным лишь для ограниченного числа лиц, знающих эту систему. Применялась для зашифровки военных, дипломатических, торгово-финансовых, нелегально-политических, религиозно-еретических текстов.».

Зашифрованная фраза:
«БББЗАИБЖВАБЕАГВИААВВАИГДГЗВАААБАБДБЕБЖАИБИГБГЖБИБЖА
ЕВДАИААБВГББДААГДГЕБИАИБИВААЕБГААГЕАИАЗБГАЕБДАЕБДАИГ
ДГЕБЕАБГАВЕБДБЕАГБЕГЕБЖАИБИГББГААГЖАИБИБЖБЕБВГБАЗВБАЕ
БГААГДГЕВДАЕБВГБГГГЕБИАДАЕБВААВАГБГЕВААЕБББИВГГЕБЖБЕБ
ДГДВАБДГАБГГЕБВАИВЖГВГЕБИАДАЕБВААВАГБГЕВААЕБББИВГГЕБ
ЖБЕБДГДВАБДГАБГГЕБВАИВЖГВГЕАДБВГДГЕБЕАГБЗААБДАИВЕАЕБД
БДБЕАГБЕГЕВЕАИБИБВААГЕБВАИВДГЖАЗБДААГГВЗАИВГГЕГВВАВБГ
ЕБИАИБИВААЕБГВБГИБЖБЗАИБГАЕБДГДБВААБИГБГЕАДБВГДГЕАЗАА
ВЖАИВВБЗБЕАВББАИГЕАВБЕАЕБДБДГАВГГЖАДАИБЖБВБЕБГААВААИ
ВЕАЕБИББАИВГГЖВАБЕБЗАГБЕАВБЕГЗВВАИБДААБДБИБЕАВГАВГГЖБ
ДАЕБВАЕАГААБВГББДБЕГЗБЖБЕБВАИВААИВЕАЕБИББАИВГГЖБЗАЕБВ
АИАГАИБЕАЗБДБЕГЗАЕБЗАЕВААИВЕАЕБИББАИВГГЕВААЕБББИВАБЕА
ВГИ».

5. Шифр Вижинера.

В данном шифре ключ задается фразой из d букв. Ключевая фраза подписывается с повторением под сообщением. Букву шифротекста необходимо находить на пересечении столбца, определяемого буквой открытого текста, и строки, определяемой буквой ключа: $Vig_d(m_i)=(m_i+k_{i \bmod d})(\text{mod } n)$, где m_i, k_i, $Vig_d(m_i)$ - порядковые номера в алфавите очередных символов открытого текста, ключа и шифротекста соответственно. Обратное преобразование выглядит следующим образом: $Vig_d^{-1}(m_i)=(m_i-k_{i \bmod d}+n)(\text{mod } n)$. Пример алфавита приведен в таблице 9 (мощность алфавита n=34). В таблице 10 записан шифротекст «сокрытие информации» с номерами букв, ключевое слово с повторением «ключ» (d=4). Каждая буква шифрованного сообщения получается суммой букв фразы и ключа/mod 34, например, первая буква 19(с)+12(к):mod34=31(э); 16(о)+13(л):mod 34=29(ы); 12(к)+32(ю):mod34=10(и) и т.д. Поэтому фраза «сокрытие информации» будет выглядеть как «эыизёяжэкхллъэкшбхж».

Таблица 9 - Пример алфавита для шифрования

1 а	2 б	3 в	4 г	5 д	6 е	7 ё	8 ж	9 з	10 и	11 й	12 к	13 л	14 м	15 н	16 о	17 п
18 р	19 с	20 т	21 у	22 ф	23 х	24 ц	25 ч	26 ш	27 щ	28 ъ	29 ы	30 ь	31 э	32 ю	33 я	34 пробел

Таблица 10 - Пример работы алгоритма шифрования.

с 19	о 16	к 12	р 18	ы 29	т 20	и 10	е 6	 34	и 10	н 15	ф 22	о 16	р 18	м 14	а 1	ц 24	и 10	и 10
к 12	л 13	ю 32	ч 25	к 12	л 13	ю 32	ч 25	к 12	л 13	ю 32	ч 25	к 12	л 13	ю 32	ч 25	к 12	л 13	ю 32
31 э	29 ы	10 и	9 з	7 ё	33 я	8 ж	31 э	12 к	23 х	13 л	13 л	28 ъ	31 э	12 к	26 ш	2 б	23 х	8 ж

Пример расчета. Фамилия и имя студентки – Толуспаева Данагуль. Каждый студент самостоятельно решает, какие символы внести в свой алфавит шифрования, а какие - убрать. В данном случае: убирается из алфавита буква «ё» и добавляется знак пробела «_» и запятая, итого мощность алфавита = 34, то есть таблица алфавита:

1 а	2 б	3 в	4 г	5 д	6 е	7 ж	8 з	9 и	10 й
11 к	12 л	13 м	14 н	15 о	16 п	17 р	18 с	19 т	20 у
21 ф	22 х	23 ц	24 ч	25 ш	26 щ	27 ъ	28 ы	29 ь	30 э
31 ю	32 я	33 _	34 ,						

Далее вписывается ключевая фраза с повторением:

к 11	р 17	и 9	п 16	т 19	о 15	г 4	р 17	а 1	ф 21	и 9	я 32	_ 33
т 19	о 15	л 12	у 20	с 18	п 16	а 1	е 6	в 3	д 5	н 14	г 4	ь 29
э 30	я 32	ф 21	б 2	в 3	ю 31	д 5	ц 23	г 4	щ 26	ц 23	б 2	ы 28

Исходная фраза: «Криптография - тайнопись, специальная система изменения обычного письма, используемая с целью сделать текст понятным лишь для ограниченного числа лиц, знающих эту систему. Применялась для зашифровки военных, дипломатических, торгово-финансовых, нелегально-политических, религиозно-еретических текстов.».

Зашифрованная фраза:
«эяфбвюдцгщцбыгпх,_яйчяд/яуазчмявэбгб\ццхгшымт/ъянлркымъс/эннзэпйсг\эмвгымуъ,рюо,хчахыйт/бо\чло,кв/вчфчфвк,\шипяцы/аэщсвэьтб/рцьчс\учср/юдцгтцыа_ьрч_о\шофров\юыгл/юяряялъм\,гдн\эьбажтц/фюмзшьйят,эд\зрлв\вуефжаюгрлг/рта_ьезсфйхоуъдгыдсгьшце/хуюз,хэкжъэбуфуряжт/ьсячубсятьвк,ъфдъежчннгг/бшъфчъючусгуфагчбщбъйыб/чуовгэо».

6. Шифр Гронсфельда.

Идея шифра Виженера (использование многоалфавитной подстановки) используется в шифре Гронсфельда. Он использует в качестве ключа целое число. Каждая цифра в десятичной записи этого числа означает величину сдвига заменяющего алфавита при подстановке соответствующей буква открытого текста. Если K – ключ, d – количество цифр в нем, а K_i – i-я десятичная цифра ключа, то шифрование можно представить следующим образом: $Gro(m_i)=(m_i+k_{i \mod d})(\mathrm{mod}\ n)$. Таким образом, шифр повторяет процедуру шифрованию Виженера, но вместо порядкового номера символа ключа в алфавите использует непосредственное десятичное значение (недостатком является уменьшении величины сдвига для каждого символа величиной 9).

Пояснение на примере. Исходная позиция букв в алфавите приведена в таблице 11. Выписываем фразу для шифрования с указанием снизу номера буквы из алфавита (таблица 11), выбираем произвольный ключ-число 14274 и подписываем его с повторением ниже, далее суммируя цифру номера буквы с соответствующей цифрой ключа и приводим в алфавитный вид. Фраза «сокрытие информации» будет выглядеть как «ттмчяумжёмошрчрбъкп» (таблица 12).

Таблица 11 - Исходная позиция букв в первоначальном алфавите

1 а	2 б	3 в	4 г	5 д	6 е	7 ё	8 ж	9 з	10 и	11 й
12 к	13 л	14 м	15 н	16 о	17 п	18 р	19 с	20 т	21 у	22 ф
23 х	24 ц	25 ч	26 ш	27 щ	28 ъ	29 ы	30 ь	31 э	32 ю	33 я
34 пробел	35 -									

Таблица 12 - Получение шифротекста

с	о	к	р	ы	т	и	е		и	н	ф	о	р	м	а	ц	и	и
19	16	12	18	29	20	10	6	34	10	15	22	16	18	14	1	24	10	10
									+									
1	4	2	7	4	1	4	2	7	4	1	4	2	7	4	1	4	2	7
									||									
20	20	14	25	33	21	14	8	7	14	16	26	18	25	18	2	28	12	17
т	т	м	ч	я	у	м	ж	ё	м	о	ш	р	ч	р	б	ъ	к	п

Пример расчета. Задумывается некоторое число в качестве кодового числа, например, 291294. Алфавит – 34 символа (нет ё, есть пробел и тире) – таблица 13.

Таблица 13 – Таблица алфавита

1 а	2 б	3 в	4 г	5 д	6 е	7 ж	8 з	9 и	10 й
11 к	12 л	13 м	14 н	15 о	16 п	17 р	18 с	19 т	20 у
21 ф	22 х	23 ц	24 ч	25 ш	26 щ	27 ъ	28 ы	29 ь	30 э
31 ю	32 я	33 _	34 -						

Процедура получения шифротекста представлен в таблице 14.

Таблица 14 – Наложение фразы и кодового числа

к	р	и	п	т	о	г	р	а	ф	и	я	-
11	17	9	16	19	15	4	17	1	21	9	32	34
+	+	+	+	+	+	+	+	+	+	+	+	+
2	9	1	2	9	4	2	9	1	2	9	4	2
13	26	10	18	28	19	6	26	2	23	18	2	2
м	щ	й	с	ы	т	е	щ	б	ц	с	б	б

Исходная фраза: «Криптография - тайнопись, специальная система изменения обычного письма, используемая с целью сделать текст понятным лишь для ограниченного числа лиц, знающих эту систему. Применялась для зашифровки военных, дипломатических, торгово-финансовых, нелегально-политических, религиозно-еретических текстов.».

Зашифрованная фраза:
«Мщйсытещбцсбб ыблцтсстюз хсочкйпюцб-з хкъузхда сиоосзцй-з тгвшпчзрз ркъ_ой- къурфэйьйой_а ъвшомюевунжнйцюз узухфз ычсюыоэхв нсщюз инж-рмфвцйщоспчдр _муфбафншзипйаысцадцхзткъцзхф сымооо-фдуг- жфба рбъсштчгмсв дчжпцячз екшпрхбфсызълкюв фчсечжрзхкцдпъпдвща цжнозвфэпчгсчмкымщотмсща ыжнсзкчипчгзыжфсызълкюв фолуыгд».

7. Шифрование биграммами.

Наиболее известный шифр биграммами называется Playfair. Пояснение на примере. Ключевая фраза без повторения букв вписывается в таблицу, далее записываются остальные неиспользованные буквы алфавита. Текст для шифровки разбивается на пары букв (биграммы), и шифрование строится по следующим трём простым правилам (таблица 15).

Таблица 15 - Шифрование текста

Ш	И	Ф	Р	О	В
А	Н	Е	Б	Г	Д
Ж	З	Й	К	Л	М
П	С	Т	У	Х	Ц
Ч	Щ	Ь	Ы	Ъ	Э
Ю	Я		.	,	-

1) Если обе буквы биграммы исходного текста принадлежат одной колонке таблицы, то буквами шифра считаются буквы, которые лежат под ними. Так, биграмма ИН дает текст шифровки НЗ. Если буква открытого текста находится в нижнем ряду, то для шифра берется соответствующая буква из верхнего ряда, и биграмма НЯ дает шифр ЗИ. (Биграмма из одной буквы или пары одинаковых букв тоже подчиняется и ОО дает ГГ).

2) Если обе буквы биграммы исходного текста принадлежат одной строке таблицы, то буквами шифра считаются буквы, которые лежат справа от них. Так, биграмма АБ дает текст шифровки НГ. Если буква открытого текста находится в крайней правой колонке, то для шифра берется буква из крайней левой колонки той же строки, и биграмма АД дает шифр НА.

3) Если обе буквы биграммы открытого текста лежат в разных рядах и колонках, то вместо них берутся такие две буквы, чтобы вся их четверка представляла прямоугольник. Например, биграмма ЕК шифруется как БЙ (КЕ зашифруется ЙБ).

Заполнение квадрата алфавитом может быть случайным, а может определяться некоторой ключевой фразой, все символы которой (но без повторений) записываются в начале матрицы, а затем по порядку выписываются остальные буквы алфавита.

Ключ – «ШИФРОВАНИЕ», текст для шифрования – «СОКРЫТИЕ ИНФОРМАЦИИ», результат будет выглядеть как «ХИУБЬУФНЯФЕИВОЖДСВРЯ».

Пример расчета. Фамилия и имя студентки – Толуспаева Данагуль. Таблица с ключом приведена в таблице 16.

Таблица 16 – Первоначальная таблица

Т	О	Л	У	С	П
А	Е	В	Д	Н	Г
Ь	Б	Ж	З	И	Й
К	М	Р	Ф	Х	Ц
Ч	Ш	Щ	Ъ	Ы	Э
Ю	Я	,		-	.

Исходная фраза: «Криптография - тайнопись, специальная система изменения обычного письма, используемая с целью сделать текст понятным лишь для ограниченного числа лиц, знающих эту систему. Применялась для зашифровки военных, дипломатических, торгово-финансовых, нелегально-политических, религиозно-еретических текстов.».

Зашифрованная фраза: «ФМ ЙС ЛО ВЦ ДК Б- ЮС ГЬ ЕС СЙ ТИ –Л ОГ ХЙ ВТ ИА ЕЮ –У НХ ОА КЕ –З БФ ГВ ГВ Б- ЯЦ ЧШ ЫА ПЕ УЯ СЙ ТИ КЕ –Ж ТП УЛ ИБ ОД КЕ -, У- МГ ТЖ –Я УН ВО АЬ ЗЮ ОА ХТ УЮ ПТ Е- СА ШХ ,У БЫ ЗЮ ВУ -, ПЕ КВ ХИ ША ИИ ПЕ УЯ ЫЬ ПУ ДЮ СЖ Р. ИД ТЬ ЫЖ Ф-ЧП УД ХН ПО ШБ П_ ЛЦ БХ ГВ ,О НТ ЗЮ ВУ -, ЬД ЫБ ХФ ЛЕ ХЬ ,Д БЕ ИИ –Ы –В ЙС УЛ КЕ СЬ ША ТХ ЫХ ЮЛ ЛМ ЕП ЕЛ _Х ИХ ГЕ ЛП НЩ Р- ВГ ОВ

ЕА ТЖ ЕС .С УЛ ЬС ЬЫ НО ХЬ Р- МВ СЖ НЙ УБ ЕС ЯН МВ СЬ ША ТХ ЫХ ЮУ АМ ОП ЛЕ Ю-».

8. Шифрование биграммами с двойным квадратом.

В 1854 году англичанин Чарльз Уитстон разработал новую шифровку биграммами, которую называют «двойной квадрат». Шифрование происходит аналогично шифру Playfair, но биграммы шифруются по двум таблицам, заполненными алфавитами (для примера в левой таблице 17 ключом может быть фамилия, а в правой – имя студента, написанные без повторения одинаковых букв). Для пары символов из исходного сообщения строится прямоугольник в двух таблицах по правилу – первая буква в левой таблице является одним углом, вторая - в правой - другим. Буквы биграммы шифра берутся из двух оставшихся вершин прямоугольника. Если обе буквы лежат в одних и тех же строках, то буквы шифра берут из той же строки, но в следующем столбце таблицы (для последнего столбца – из первого столбца). Текст для шифрования – «СОКРЫТИЕ ИНФОРМАЦИИ.», результат будет выглядеть как «ФЗНСШХЯДЮАОУПСИБТЛЯ,».

Таблица 17 - Начальные таблицы для шифрования

Ф	А	М	И	Л	Я
Б	В	Г	Д	Е	Ж
З	К	Н	О	П	Р
С	Т	У	Х	Ц	Ч
Ш	Щ	Ь	Ы	Ъ	Э
Ю		.	,	-	!

И	М	Я	А	Б	В
Г	Д	Е	Ж	З	К
Л	Н	О	П	Р	С
Т	У	Ф	Х	Ц	Ч
Ш	Щ	Ь	Ы	Ъ	Э
Ю		.	,	-	!

Пример расчета. Фамилия и имя студентки – Толуспаева Данагуль. В таблице 18 дано представление с ключом.

Таблица 18 - Две исходные таблицы для получения шифра

Т	О	Л	У	С	П
А	Е	В	Б	Г	Д
Ж	З	И	Й	К	М
Н	Р	Ф	Х	Ц	Ч
Ш	Щ	Ъ	Ы	Ь	Э
Ю	Я		,	-	.

Д	А	Н	Г	У	Л
Ь	Б	В	Е	Ж	З
И	Й	К	М	О	П
Р	С	Т	Ф	Х	Ц
Ч	Ш	Щ	Ъ	Ы	Э
Ю	Я	-	,		.

Исходная фраза: «Криптография - тайнопись, специальная система изменения обычного письма, используемая с целью сделать текст понятным лишь для ограниченного числа лиц, знающих эту систему. Применялась для зашифровки военных, дипломатических, торгово-финансовых, нелегально-политических, религиозно-еретических текстов.», разбивка: «Кр_ип_то_гр_аф_ия_ -_ т_ай_но_пи_сь_, _сп_ец_иа_ль_на_я _си_ст_ем_а _из_ме_не_ни_я _об_ыч_но_го_ п_ис_ьм_а,_ и_сп_ол_ьз_уе_ма_я _с _це_ль_ю _сд_ел_ат_ь _те_кс_т _по_ня_тн_ым_ л_иш_ь _дл_я _ог_ра_ни_че_нн_ог_о _чи_сл_а _ли_ц,_ з_на_ющ_их_ э_ту_ с_ис_те_му_. _Пр_им_ен_ял_ас_ь _дл_я _за_ши_фр_ов_ки_ в_ое_нн_ых_, _ди_пл_ом_ат_ич_ес_ки_х,_ т_ор_го_во_-

ф_ин_ан_со_вы_х,_ н_ел_ег_ал_ьн_о-_по_ли_ти_че_ск_их_, _ре_ли_ги_оз_но_-е_ре_ти_че_ск_их_ т_ек_ст_ов_.».

Зашифрованная фраза: «ИЦ КМ УЖ ЬЦ ЕН Й_ -Ц БЖ ХЖ ДМ ДГ ЯХ ГД РК ЗТ ЩЛ БЮ ЯФ ЙФ ГА ЙП .В ЙЙ ВО ВО Й_ _И ЖЫ ТП АУ УЯ ДМ ДГ ЙП ЮЙ ЛК АП ЭВ ГБ ЙП Я- ЛК ФГ ДВ Ю- УТ ЗО ВН Ы_ ГА ЙЦ УЮ УМ СЮ ДЛ ЪЙ .Л ЙЬ Ы_ ЗП Я- АУ СО РЖ ФД ТТ АУ УЯ РМ УП ЖЮ ДИ Ф- КО ЬЮ ЧЗ Х, ЩЧ У, ДК НЦ ЕЗ Л, ДЧ КЙ ВО .О БН Ы_ ЗП Я- ЙО ЧЖ ТН НЕ ОЖ –В ГЕ ТТ ЫХ ЮУ КМ УИ ЙП ДЖ ФД НК ОФ –Х ДР ЖК ЖИ ,Ц КЛ ВТ УК ЖЬ Ф, ФА ГВ БС ДВ ХЖ .К АП КФ ИЬ БР ОЖ Ф, ФЕ ДИ ЬК ЛЕ ХЖ ,Г ФЕ ДЖ ФД НК ОФ –Ф ВЗ НЦ НЕ -.»

9. Книжные шифры квадрата Кардано.

В XVI веке итальянский математик и философ Дж. Кардано предложил новый тип шифра, основанный на очень простой и в то же время надежной перестановке букв открытого текста. Для шифрования он предложил использовать квадрат с прорезанными в нем несколькими ячейками. Ячейки прорезались таким образом, чтобы при повороте квадрата на 90, 180 и 270 градусов в прорезях поочередно появлялись все позиции исходного квадрата, причем по одному разу. Шифр получил название *квадрата Кардано,* примеры квадратов 4х4 и 5х5 приведены на рисунке 2.

При шифровании квадрат накладывается на лист бумаги сначала в исходном положении, и в прорези записывается первая часть сообщения, затем квадрат поворачивается на 90º, и в прорези вписывается вторая часть, и т.д. После того как будут заполнены все ячейки квадрата, шифротекст считывается их него построчно.

Пример осуществления шифрования «СОКРЫТИЕ ИНФОРМАЦИИ» представлен на рисунке 3: «ЫОС РТИОНКМИРФЕАЦИИ».

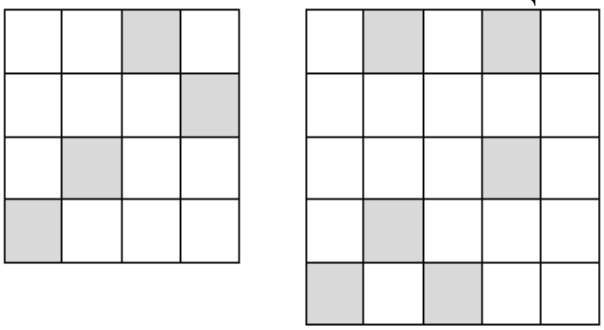


Рисунок 2 – Пример квадратов Кардано: 4·4 и 5·5

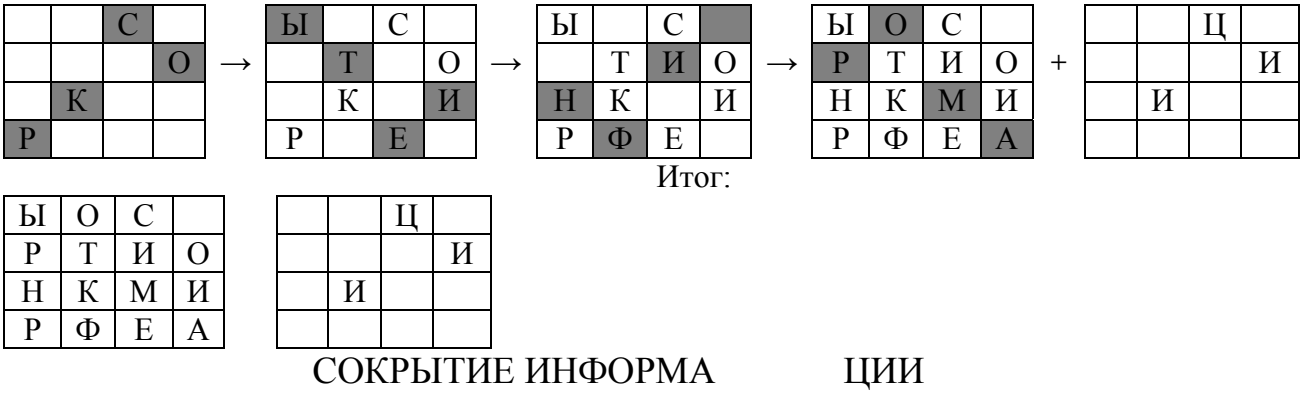
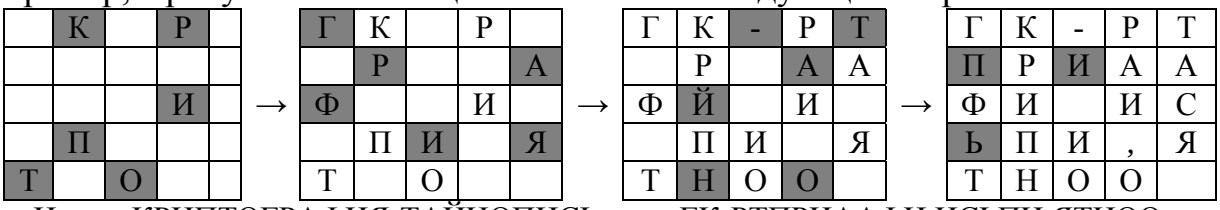


Рисунок 3 - Итерации получения шифрования с помощью таблиц Кардано

Пример расчета. Произвольно выбираю форму в прорезях квадрата, например, с рисунка 2. Таблицы заполняются следующим образом:



	К		Р	
			И	
	П			
Т		О		

→

Г	К		Р	
	Р			А
Ф			И	
	П	И		Я
Т		О		

→

Г	К	-	Р	Т
	Р		А	А
Ф	Й		И	
	П	И		Я
Т	Н	О	О	

→

Г	К	-	Р	Т
П	Р	И	А	А
Ф	И		И	С
Ь	П	И	,	Я
Т	Н	О	О	

Итог: «КРИПТОГРАФИЯ-ТАЙНОПИСЬ, » = «ГК-РТПРИААФИ ИСЬПИ,ЯТНОО »

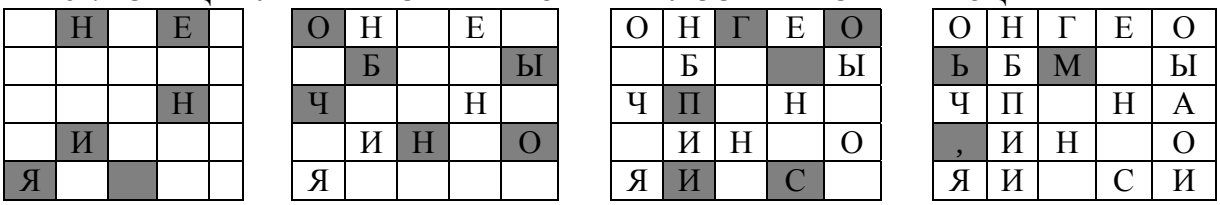
	С		П	
			Е	
	Ц			
И		А		

Л	С		П	
	Ь			Н
А			Е	
	Ц	Я		
И		А		

Л	С	С	П	И
	Ь		С	Н
А	Т		Е	
	Ц	Я		
И	Е	А	М	

Л	С	С	П	И
А	Ь		С	Н
А	Т		Е	И
З	Ц	Я	М	
И	Е	А	М	Е

Итог: «СПЕЦИАЛЬНАЯ ИСТЕМА ИЗМЕ» = «ЛССПИАЬ СНАТ ЕИЗЦЯМ ИЕАМЕ»



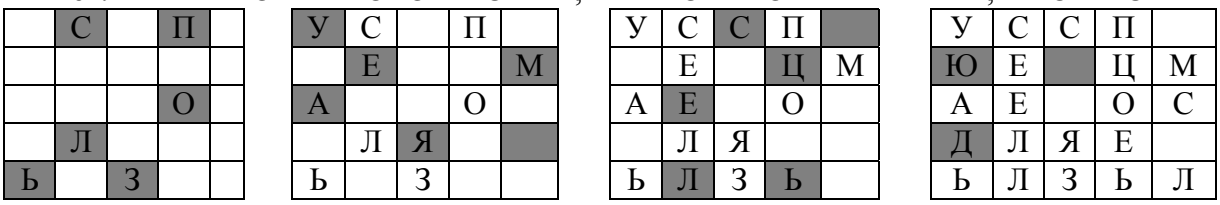
	Н		Е	
			Н	
	И			
Я				

О	Н		Е	
	Б			Ы
Ч			Н	
	И	Н		О
Я				

О	Н	Г	Е	О
	Б			Ы
Ч	П		Н	
	И	Н		О
Я	И		С	

О	Н	Г	Е	О
Ь	Б	М		Ы
Ч	П		Н	А
,	И	Н		О
Я	И		С	И

Итог: «НЕНИЯ ОБЫЧНОГО ПИСЬМА, И» = «ОНГЕОЬБМ ЫЧП НА,ИН ОЯИ СИ»



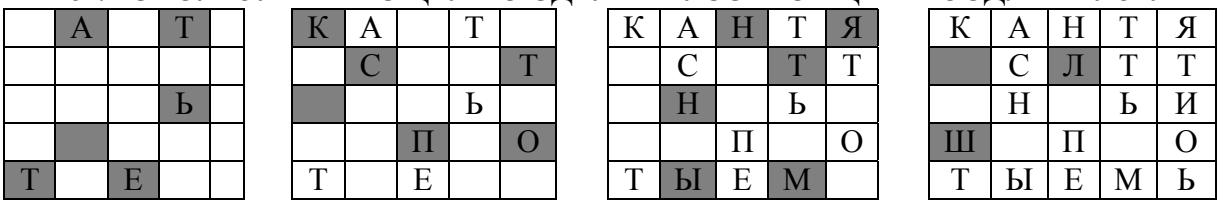
	С		П	
			О	
	Л			
Ь		З		

У	С		П	
	Е			М
А			О	
	Л	Я		
Ь		З		

У	С	С	П	
	Е		Ц	М
А	Е		О	
	Л	Я		
Ь	Л	З	Ь	

У	С	С	П	
Ю	Е		Ц	М
А	Е		О	С
Д	Л	Я	Е	
Ь	Л	З	Ь	Л

Итог: «СПОЛЬЗУЕМАЯ С ЦЕЛЬЮ СДЕЛ» = «УССП ЮЕ ЦМАЕ ОСДЛЯЕ ЬЛЗЬЛ»



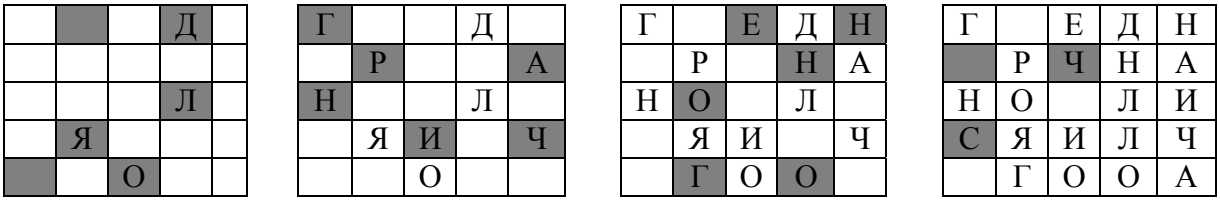
	А		Т	
			Ь	
Т		Е		

К	А		Т	
	С			Т
			Ь	
		П		О
Т		Е		

К	А	Н	Т	Я
	С		Т	Т
	Н		Ь	
		П		О
Т	Ы	Е	М	

К	А	Н	Т	Я
	С	Л	Т	Т
	Н		Ь	И
Ш		П		О
Т	Ы	Е	М	Ь

Итог: «АТЬ ТЕКСТ ПОНЯТНЫМ ЛИШЬ» = «КАНТЯ СЛТТ Н ЬИШ П ОТЫЕМЬ»



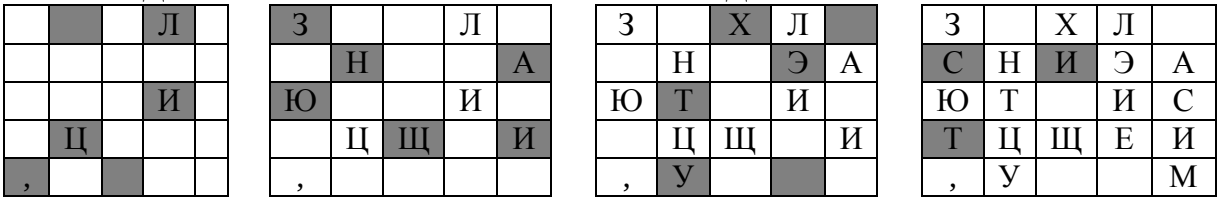
			Д	
			Л	
	Я			
		О		

Г			Д	
	Р			А
Н			Л	
	Я	И		Ч
		О		

Г		Е	Д	Н
	Р		Н	А
Н	О		Л	
	Я	И		Ч
	Г	О	О	

Г		Е	Д	Н
	Р	Ч	Н	А
Н	О		Л	И
С	Я	И	Л	Ч
	Г	О	О	А

Итог: « ДЛЯ ОГРАНИЧЕННОГО ЧИСЛА» = «Г ЕДН РЧНАНО ЛИСЯИЛЧ ГООА»



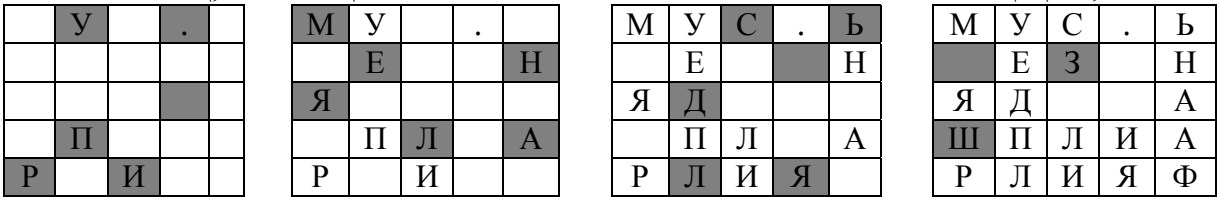
			Л	
			И	
	Ц			
,				

З			Л	
	Н			А
Ю			И	
	Ц	Щ		И
,				

З		Х	Л	
	Н		Э	А
Ю	Т		И	
	Ц	Щ		И
,	У			

З		Х	Л	
С	Н	И	Э	А
Ю	Т		И	С
Т	Ц	Щ	Е	И
,	У			М

Итог: « ЛИЦ, ЗНАЮЩИХ ЭТУ СИСТЕМ» = «З ХЛ СНИЭАЮТ ИСТЦЩЕИ,У М»



	У		.	
	П			
Р		И		

М	У		.	
	Е			Н
Я				
	П	Л		А
Р		И		

М	У	С	.	Ь
	Е			Н
Я	Д			
	П	Л		А
Р	Л	И	Я	

М	У	С	.	Ь
	Е	З		Н
Я	Д			А
Ш	П	Л	И	А
Р	Л	И	Я	Ф

Итог: «У. ПРИМЕНЯЛАСЬ ДЛЯ ЗАШИФ» = «МУС.Ь ЕЗ НЯД АШПЛИАРЛИЯФ»

Исходная фраза: «Криптография - тайнопись, специальная система изменения обычного письма, используемая с целью сделать текст понятным лишь для ограниченного числа лиц, знающих эту систему. Применялась для зашиф».

Зашифрованная фраза: «ГК-РТПРИААФИ ИСЬПИ,ЯТНОЛССПИАЬ СНАТ ЕИЗЦЯМ ИЕАМЕОНГЕОЬБМ ЫЧП НА,ИН ОЯИ СИУССП ЮЕ ЦМАЕ ОСДЛЯЕ ЬЛЗЬЛКАНТЯ СЛТТ Н ЬИШ П ОТЫЕМЬГ ЕДН РЧНАНО ЛИСЯИЛЧ ГООАЗ ХЛ СНИЭАЮТ ИСТЦЩЕИ,У ММУС.Ь ЕЗ НЯД АШПЛИАРЛИЯФ».

10. Перестановочный шифр с ключевым словом.

Буквы ключевого слова без повторений записываются в первую строку таблицы, определяя таким образом количество ее столбцов. Буквы сообщения записываются в таблицу построчно. Сформированная таким образом таблица сортируется по столбцам, критерием сортировки является порядок следования символа первой строки в алфавите. После сортировки шифрованный текст переписывается по столбцам.

Пояснение на примере. Ключ: «ФАМИЛИЯИМЯСТУДЕНТА» (без повторения «ФАМИЛЯСТУДЕН»). Открытый текст: «СОКРЫТИЕ ИНФОРМАЦИИ» (таблица 19).

Таблица 19 - Получение шифра

Ф	А	М	И	Л	Я	С	Т	У	Д	Е	Н
С	О	К	Р	Ы	Т	И	Е		И	Н	Ф
О	Р	М	А	Ц	И	И	-	-	-	-	-

→

А	Д	Е	И	Л	М	Н	С				
О	И	Н	Р	Ы	К	Ф	И	Е		С	Т
Р	-	-	А	Ц	М	-	И	-	-	О	И

Шифротекст: «ОРИ-Н-РАЫЦКМФ-ИИЕ- -СОТИ»

Пример расчета. Фамилия и имя студентки – Толуспаева Данагуль. В таблице 20 представлен полученный расклад.

Исходная фраза: «КРИПТОГРАФИЯ - ТАЙНОПИСЬ, СПЕЦИАЛЬНАЯ СИСТЕМА ИЗМЕНЕНИЯ ОБЫЧНОГО ПИСЬМА, ИСПОЛЬЗУЕМАЯ С ЦЕЛЬЮ СДЕЛАТЬ ТЕКСТ ПОНЯТНЫМ ЛИШЬ ДЛЯ ОГРАНИЧЕННОГО ЧИСЛА ЛИЦ, ЗНАЮЩИХ ЭТУ СИСТЕМУ. ПРИМЕНЯЛАСЬ ДЛЯ ЗАШИФРОВКИ ВОЕННЫХ, ДИПЛОМАТИЧЕСКИХ, ТОРГОВО-ФИНАНСОВЫХ, НЕЛЕГАЛЬНО-ПОЛИТИЧЕСКИХ, РЕЛИГИОЗНО-ЕРЕТИЧЕСКИХ ТЕКСТОВ.».

Зашифрованная фраза:
«СЯСЛТНЧЬЬ ЕСЫЛЧИЗТУЛЗИ,ИОННОЕИРХ.П-ЬЬЕЕНМЗЦЛТМЯЕСНУ-АА ДЧРАЕ-СГЕ –ОАСАААИГ,ЕЛТПЛОНАЮСРЬИОПСОСЕОИОИЕ-ГЙПЯ ЯОИМЬЬОИГО ЩИИ ФЕЛКВОГЛХЗЧК-РИИАСЕЫСЛСДКНДИЧ,ЭМЯ КХТТИ,НЧЛЕИВКФПКФПИИМБИО СЕТ Н Ц Е НЯВЫА,ФХЬИЕ-КОТТ, НМНОАУЕА НЛА ПСШВИЕГНЛПКИТТ-АОЦСЗОППЯ ТЯЬАОИХТЕЛОНМ Х-ЫЛТРОСТ-РНЕ И САЮ НШРГЛИСМДРНОИОВАИ,НЕС-».

Таблица 20 - Расчет шифрования

Т	О	Л	У	С	П	А	Е	В	→	Л	У	П	А	О	Т	С	В	Е
К	Р	И	П	Т	О	Г	Р	А		С	П	О	Г	Р	К	Т	А	Р
Ф	И	Я	-	Т	А	Й	Н	О		Я	-	А	Й	И	Ф	Т	О	Н
П	И	С	Ь	,	С	П	Е	Ц		С	Ь	С	П	И	П	,	Ц	Е
И	А	Л	Ь	Н	А	Я		С		Л	Ь	А	Я	А	И	Н	С	
И	С	Т	Е	М	А		И	З		Т	Е	А		С	И	М	З	И
М	Е	Н	Е	Н	И	Я		О		Н	Е	И	Я	Е	М	Н	О	
Б	Ы	Ч	Н	О	Г	О		П		Ч	Н	Г	О	Ы	Б	О	П	
И	С	Ь	М	А	,	И	С	П		Ь	М	,	И	С	И	А	П	С
О	Л	Ь	З	У	Е	М	А	Я		Ь	З	Е	М	Л	О	У	Я	А
	С		Ц	Е	Л	Ь	Ю				Ц	Л	Ь	С		Е		Ю
С	Д	Е	Л	А	Т	Ь		Т		Е	Л	Т	Ь	Д	С	А	Т	
Е	К	С	Т		П	О	Н	Я		С	Т	П	О	К	Е		Я	Н
Т	Н	Ы	М		Л	И	Ш	Ь		Ы	М	Л	И	Н	Т		Ь	Ш
	Д	Л	Я		О	Г	Р	А		Л	Я	О	Г	Д			А	Р
Н	И	Ч	Е	Н	Н	О	Г	О		Ч	Е	Н	О	И	Н	Н	О	Г
	Ч	И	С	Л	А		Л	И		И	С	А		Ч		Л	И	Л
Ц	,	З	Н	А	Ю	Щ	И	Х		З	Н	Ю	Щ	,	Ц	А	Х	И
	Э	Т	У		С	И	С	Т		Т	У	С	И	Э			Т	С
Е	М	У	.	П	Р	И	М	Е		У	.	Р	И	М	Е	П	Е	М
Н	Я	Л	А	С	Ь		Д	Л		Л	А	Ь		Я	Н	С	Л	Д
Я		З	А	Ш	И	Ф	Р	О		З	А	И	Ф		Я	Ш	О	Р
В	К	И		В	О	Е	Н	Н		И		О	Е	К	В	В	Н	Н
Ы	Х	,	Д	И	П	Л	О	М		,	Д	П	Л	Х	Ы	И	М	О
А	Т	И	Ч	Е	С	К	И	Х		И	Ч	С	К	Т	А	Е	Х	И
,	Т	О	Р	Г	О	В	О	-		О	Р	О	В	Т	,	Г	-	О
Ф	И	Н	А	Н	С	О	В	Ы		Н	А	С	О	И	Ф	Н	Ы	В
Х	,	Н	Е	Л	Е	Г	А	Л		Н	Е	Е	Г	,	Х	Л	Л	А
Ь	Н	О	-	П	О	Л	И	Т		О	-	О	Л	Н	Ь	П	Т	И
И	Ч	Е	С	К	И	Х	,	Р		Е	С	И	Х	Ч	И	К	Р	,
Е	Л	И	Г	И	О	З	Н	О		И	Г	О	З	Л	Е	И	О	Н
-	Е	Р	Е	Т	И	Ч	Е	С		Р	Е	И	Ч	Е	-	Т	С	Е
К	И	Х		Т	Е	К	С	Т		Х		Е	К	И	К	Т	Т	С
О	В	.	-	-	-	-	-	-		.	-	-	-	В	О	-	-	-

2 Расчетно-графическая работа №2. Алгоритмы сортировки

Цель работы: приобретение знаний по работе разных алгоритмов сортировок, знание критериев оценки работы алгоритмов сортировок.

2.1 Рабочее задание

Дается массив данных по вариантам, его надо отсортировать по возрастанию всеми описанными алгоритмами сортировки и сделать оценку производительности их работы:

1) Пузырьковая сортировка.
2) Шейкер-сортировка.
3) Сортировка посредством выбора.
4) Сортировка вставками.
5) Сортировка Шелла.
6) Быстрая.

Вариант 1.
1 -22 66 2 5 78 -6 -6 -4 0 11 3 36 2
Вариант 2.
2 3 -3 0 0 8 9 9 11 78 -9 -7 -8 -55
Вариант 3.
3 2 1 3 0 -8 -7 -8 -44 89 -45 7 12 -12
Вариант 4.
22 11 1 0 22 -4 -4 45 9 96 66 7 54 53
Вариант 5.
9 8 3 5 7 -2 -6 5 -6 11 -12 -78 -8 88
Вариант 6.
23 -32 -1 -2 -3 -4 -6 -4 -7 -8 -8 -9 0 8
Вариант 7.
5 -6 -5 -5 1 2 0 -4 44 -55 22 1 100 8
Вариант 8.
6 3 2 7 2 1 10 22 8 -5 -6 -56 101 0
Вариант 9.
34 54 67 32 98 90 23 87 -54 -63 -32 -43 51 -54
Вариант 10.
12 23 34 45 -12 -34 -43 0 -6 -8 78 -52 52 11
Вариант 11.
22 12 -9 -7 12 -36 -96 106 -52 -42 400 896 -4 0
Вариант 12.
16 25 -14 -25 5 12 -78 -67 20 47 45 47 6 -9
Вариант 13.
30 78 -52 -9 -50 -74 -90 25 107 702 1 2 23 -8
Вариант 14.
89 67 -43 -5 2 -2 78 34 -9 -6 -78 -4 -4 12

Вариант 15.

18 16 17 2 -4 8 -9 1 -66 -14 -12 25 27 8

Вариант 16.

10 8 4 12 14 7 177 102 209 97 14 2 0 4

Вариант 17.

13 -26 45 -55 -9 11 -78 12 12 0 0 6 18 -99

Вариант 18.

103 -95 -45 99 300 25 458 777 -67 -90 23 -9 888 1

Вариант 19.

90 -80 110 70 -96 -1 -2 -3 4 5 8 -96 45 -89

Вариант 20.

78 12 48 -9 -66 -36 -12 -12 -25 600 23 230 56 89

Вариант 21.

1 -9 -8 -5 1 8 78 88 -66 -33 0 45 25 28

Вариант 22.

10 -22 78 -89 39 33 14 -15 -16 17 45 2 -9 -3

Вариант 23.

18 42 -99 333 17 -23 99 66 23 14 13 12 -11 2

Вариант 24.

200 18 -9 15 16 17 -18 67 688 400 -19 50 85 83

Вариант 25.

700 600 15 -56 200 -80 -60 444 666 555 -77 -88 999 500

2.2 Методические указания к выполнению работы

Алгоритм сортировки - это алгоритм для упорядочения элементов в списке по возрастанию или убыванию. Существует много различных алгоритмов сортировок. Все они имеют свои положительные стороны, общие критерии оценки алгоритма сортировки:

1) насколько быстро алгоритм сортирует информацию в среднем;

2) насколько быстро он работает в лучшем и худшем случаях;

3) естественно или неестественно он себя ведет;

4) переставляет ли он элементы с одинаковыми ключами.

Рассмотрим эти критерии.

1) Очевидно, что скорость работы любого алгоритма сортировки имеет большое значение. Скорость сортировки массива непосредственно связана с количеством сравнений и количеством обменов, происходящих во время сортировки, причем обмены занимают больше времени. Сравнение происходит тогда, когда один элемент массива сравнивается с другим; обмен происходит тогда, когда два элемента меняются местами. Время работы одних алгоритмов сортировки растет экспоненциально, а время работы других логарифмически зависит от количества элементов.

2) Время работы в лучшем и худшем случаях имеет значение, если одна из этих ситуаций будет встречаться довольно часто. Алгоритм сортировки зачастую имеет хорошее среднее время выполнения, но в худшем случае он работает очень медленно.

3) Поведение алгоритма сортировки называется естественным, если время сортировки минимально для уже упорядоченного списка элементов, увеличивается по мере возрастания степени неупорядоченности списка и максимально, когда элементы списка расположены в обратном порядке. Объем работы алгоритма оценивается количеством производимых сравнений и обменов. Если в отсортированном массиве элементы с одинаковыми ключами идут в том же порядке, в котором они располагались в исходном массиве, то алгоритм сортировки называется устойчивым, а в противном случае - неустойчивым.

4) Чтобы понять, почему переупорядочивание элементов с одинаковыми ключами имеет определенное значение, представьте себе базу данных почтовой рассылки, упорядоченную по главному ключу и подключу. Главным ключом является почтовый индекс, а в пределах одного почтового индекса записи упорядочены по фамилии. При добавлении в список нового адреса и пересортировке списка порядок подключей (то есть фамилий внутри почтовых индексов) не должен меняться. Для гарантии, что это не произойдет, алгоритм сортировки не должен обменивать ключи с одинаковым значением.

Далее представлены характерные для каждой группы алгоритмы.

1. Пузырьковая сортировка.

Этот алгоритм простейший, но эффективен он для небольших массивов. Алгоритм считается учебным и практически не применяется вне учебной литературы, вместо него на практике применяются более эффективные алгоритмы сортировки. В то же время метод сортировки обменами лежит в основе некоторых более совершенных алгоритмов таких, как шейкерная сортировка, пирамидальная сортировка и быстрая сортировка. Элементы ведут себя подобно пузырькам воздуха в воде - каждый поднимается на свой уровень.

Объяснение на примере: дан массив с числами «5 1 4 2 8», надо отсортировать его по возрастанию.

Первый проход:

(5 1 4 2 8) (1 5 4 2 8) - сравнивает два первых элемента и меняет их местами;
(1 5 4 2 8) (1 4 5 2 8) - меняет местами 5 и 4 так как 5>4;
(1 4 5 2 8) (1 4 2 5 8) - меняет местами 2 и 5 так как 5>2;
(1 4 2 5 8) (1 4 2 5 8) - стоят на своих местах (8>5), не меняет местами 5 и 8.

Второй проход:

(1 4 2 5 8) (1 4 2 5 8) - пара чисел 1 и 4 на своих местах;
(1 4 2 5 8) (1 2 4 5 8) - меняет местами 2 и 4 так как 4>2;
(1 2 4 5 8) (1 2 4 5 8) - числа 4 и 5 стоят на своих местах;
(1 2 4 5 8) (1 2 4 5 8) - числа 5 и 8 стоят в верных позициях.

Массив полностью отсортирован, но алгоритм не знает, так ли это. Поэтому ему необходимо сделать еще один полный проход и определить, что перестановок элементов не было. По отработке третьего прохода массив отсортирован, и алгоритм может быть завершён.

При анализе любого алгоритма сортировки полезно знать, сколько операций сравнения и обмена будет выполнено в лучшем, среднем и худшем случаях. Поскольку характеристики выполняемого кода зависят от таких

факторов, как оптимизация, производимая компилятором, различия между процессорами и особенности реализации, мы не будем пытаться получить точные значения этих параметров. Вместо этого сконцентрируем внимание на общей эффективности каждого алгоритма.

В пузырьковой сортировке количество сравнений всегда одно и то же, поскольку два цикла повторяются указанное количество раз независимо от того, был список изначально упорядочен или нет. Если n - количество сортируемых элементов, то внешний цикл выполняется n-1 раз, а внутренний выполняется в среднем n/2 раз, их произведение равно $(n^2-n)/2$ – это количество сравнений, которых алгоритм пузырьковой сортировки всегда выполняет.

Говорят, что пузырьковая сортировка является алгоритмом порядка n^2, поскольку время ее выполнения пропорционально квадрату количества сортируемых элементов. В алгоритме пузырьковой сортировки количество обменов в лучшем случае равно нулю, если массив уже отсортирован. Однако в среднем и худшем случаях количество обменов также является величиной порядка n^2. Есть даже два алгоритма пузырьковой сортировки: сортировка пузырьковым включением и сортировка пузырьковой выборкой, эффективность обоих одинакова.

На рисунке 4 отражен программный ход рассуждений при сортировке списка значений.

```
Сортировка 'пузырьком:'
первоначально: [-5, 1, 0, -3, 8, -2, -6, -11]
ход сортировки пошагово:
[-5, 1, 0, -3, 8, -2, -6, -11] ->
[-5, 1, 0, -3, 8, -2, -6, -11] ->
[-5, 0, 1, -3, 8, -2, -6, -11] ->
[-5, 0, -3, 1, 8, -2, -6, -11] ->
[-5, 0, -3, 1, 8, -2, -6, -11] ->
[-5, 0, -3, 1, -2, 8, -6, -11] ->
[-5, 0, -3, 1, -2, -6, 8, -11] ->
[-5, 0, -3, 1, -2, -6, -11, 8] ->
[-5, 0, -3, 1, -2, -6, -11, 8] ->
[-5, -3, 0, 1, -2, -6, -11, 8] ->
[-5, -3, 0, 1, -2, -6, -11, 8] ->
[-5, -3, 0, -2, 1, -6, -11, 8] ->
[-5, -3, 0, -2, -6, 1, -11, 8] ->
[-5, -3, 0, -2, -6, -11, 1, 8] ->
[-5, -3, 0, -2, -6, -11, 1, 8] ->
[-5, -3, 0, -2, -6, -11, 1, 8] ->
[-5, -3, -2, 0, -6, -11, 1, 8] ->
[-5, -3, -2, -6, 0, -11, 1, 8] ->
[-5, -3, -2, -6, -11, 0, 1, 8] ->
[-5, -3, -2, -6, -11, 0, 1, 8] ->
[-5, -3, -2, -6, -11, 0, 1, 8] ->
[-5, -3, -6, -2, -11, 0, 1, 8] ->
[-5, -3, -6, -11, -2, 0, 1, 8] ->
[-5, -3, -6, -11, -2, 0, 1, 8] ->
[-5, -6, -3, -11, -2, 0, 1, 8] ->
[-5, -6, -11, -3, -2, 0, 1, 8] ->
[-6, -5, -11, -3, -2, 0, 1, 8] ->
[-6, -11, -5, -3, -2, 0, 1, 8] ->
сортированый: [-11, -6, -5, -3, -2, 0, 1, 8]
```

```
программа 'сортировки пузырьком'
первоначально: [5, 1, 4, 2, 8]
ход сортировки пошагово ->
[5, 1, 4, 2, 8]
[1, 5, 4, 2, 8]
[1, 4, 5, 2, 8]
[1, 4, 2, 5, 8]
[1, 4, 2, 5, 8]
[1, 4, 2, 5, 8]
[1, 2, 4, 5, 8]
[1, 2, 4, 5, 8]
[1, 2, 4, 5, 8]
[1, 2, 4, 5, 8]
отсортировано: [1, 2, 4, 5, 8]
```

Рисунок 4 – Образец сортировки некоторого списка значений

2. Шейкер-сортировка.

Алгоритм пузырьковой сортировки можно немного улучшить, если попытаться повысить скорость его работы. Вместо того чтобы постоянно просматривать массив в одном направлении, можно чередовать направления - элементы, сильно удаленные от своих положений, быстро станут на свои места. Данная версия пузырьковой сортировки носит название шейкер-сортировки (shaker sort), поскольку действия, производимые ею с массивом, напоминают взбалтывание или встряхивание. Хотя шейкер-сортировка и является улучшенным вариантом по сравнению с пузырьковой сортировкой, она имеет время выполнения порядка n^2. Это объясняется тем, что количество сравнений не изменилось, а количество обменов уменьшилось лишь на относительно небольшую константу. Простая форма алгоритма массива после каждого прохода шейкер-сортировки:

Объяснение на примере: дан массив с числами «5 1 4 2 8», надо отсортировать его по возрастанию:

(5 1 4 2 8) (1 5 4 2 8) – ищет минимальный элемент и ставит первым;
(1 5 4 2 8) (1 5 4 2 8) – ищет максимальный элемент, ставит последним;
(1 5 4 2 8) (1 2 5 4 8) – ищет минимальное из оставшегося и ставит вторым;
(1 2 5 4 8) (1 2 4 5 8) – максимальный из оставшегося, ставит предпоследним.

3. Сортировка посредством выбора.

Она называется также сортировкой выбором или сортировкой выборками - из массива выбирается элемент с наименьшим значением и обменивается с первым элементом. Затем из оставшихся n-1 элементов снова выбирается элемент с наименьшим ключом и обменивается со вторым элементом и т.д. Эти обмены продолжаются до двух последних элементов.

Объяснение на примере: дан массив с числами «5 1 4 2 8», надо отсортировать его по возрастанию:

(5 1 4 2 8) (1 5 4 2 8) – ищет минимальный элемент и тот меняется с первым;
(1 5 4 2 8) (1 2 4 5 8) – минимальный элемент оставшегося меняет со вторым;
(1 2 4 5 8) (1 2 4 5 8) – минимальный элемент оставшегося меняет с третьим;
(1 2 4 5 8) (1 2 4 5 8) – минимальный элемент оставшегося меняет с четвертым.

К сожалению, как и в пузырьковой сортировке, внешний цикл выполняется n-1 раз, а внутренний - в среднем n/2 раз. Следовательно, сортировка посредством выбора требует $1/2(n^2-n)$ сравнений. Таким образом, это алгоритм порядка n^2, из-за чего он считается слишком медленным для сортировки большого количества элементов. Несмотря на то что количество сравнений в пузырьковой сортировке и сортировке посредством выбора одинаковое, в последней количество обменов в среднем случае намного меньше, чем в пузырьковой сортировке.

На рисунке 5 отражен ход рассуждений при сортировке списка значений.

```
Это программа 'сортировки выбором'
дан первоначальный список [2, 4, 8, 1, 0, 3, 9, 5, 7, 6]
ход сортировки пошагово ->
[2, 4, 8, 1, 0, 3, 9, 5, 7, 6]
[2, 4, 8, 1, 0, 3, 9, 5, 7, 6]
[0, 4, 8, 1, 2, 3, 9, 5, 7, 6]
[0, 1, 8, 4, 2, 3, 9, 5, 7, 6]
[0, 1, 8, 4, 2, 3, 9, 5, 7, 6]
[0, 1, 2, 4, 8, 3, 9, 5, 7, 6]
[0, 1, 2, 3, 8, 4, 9, 5, 7, 6]
[0, 1, 2, 3, 4, 8, 9, 5, 7, 6]
[0, 1, 2, 3, 4, 5, 9, 8, 7, 6]
[0, 1, 2, 3, 4, 5, 9, 8, 7, 6]
[0, 1, 2, 3, 4, 5, 9, 8, 7, 6]
[0, 1, 2, 3, 4, 5, 6, 8, 7, 9]
отсортированный список: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
Программа 'сортировки выбором'
первоначальный список [5, 1, 4, 2, 8]
ход сортировки пошагово ->
[5, 1, 4, 2, 8]
[1, 5, 4, 2, 8]
[1, 5, 4, 2, 8]
отсортированный список: [1, 2, 4, 5, 8]
```

Рисунок 5 – Два примера сортировки выбором

4. Сортировка вставками.

Сортировка вставками - сначала он сортирует два первых элемента массива. Затем алгоритм вставляет третий элемент в соответствующую порядку позицию по отношению к первым двум элементам. После этого он вставляет четвертый элемент в список из трех элементов. Этот процесс повторяется до тех пор, пока не будут вставлены все элементы.

Объяснение на примере: дан массив с числами «5 1 4 2 8», надо отсортировать его по возрастанию:

(5 1 4 2 8) (1 5 4 2 8) – сортирует два первых элемента;
(1 5 4 2 8) (1 4 5 2 8) – вставляет третий элемент по отношению к первым двум;
(1 4 5 2 8) (1 2 4 5 8) – вставляет четвертый элемент в список из трех элементов;
(1 4 5 2 8) (1 2 4 5 8) – вставляет пятый элемент в список из четырех элементов.

В отличие от пузырьковой сортировки и сортировки посредством выбора, количество сравнений в сортировке вставками зависит от изначальной упорядоченности списка. Если список уже отсортирован, количество сравнений равно n-1; в противном случае его производительность величина порядка n^2.

Вообще говоря, в худших случаях сортировка вставками настолько же плоха, как и пузырьковая сортировка и сортировка посредством выбора, а в среднем она лишь немного лучше, но у сортировки вставками есть два преимущества: ее поведение естественно (она работает меньше всего, когда массив уже упорядочен, и больше всего, когда массив отсортирован в обратном порядке, поэтому сортировка вставками - идеальный алгоритм для почти упорядоченных списков); данный алгоритм не меняет порядок одинаковых ключей (если список отсортирован по двум ключам, то после сортировки вставками он останется упорядоченным по обоим). Несмотря на то, что количество сравнений при определенных наборах данных может быть довольно низким, при каждой вставке элемента на свое место массив необходимо сдвигать. Вследствие этого количество перемещений может быть значительным.

Вышеописанные алгоритмы имеют один фатальный недостаток - время их выполнения имеет порядок n^2. Это делает сортировку больших объемов данных очень медленной, в какой-то момент эти алгоритмы становятся слишком медленными, чтобы их применять.

На рисунке 6 отражен ход рассуждений при сортировке списка значений.

```
Сортировка 'вставками:'
первоначально: [-6, -7, -10, -11, 0, -65, -32]
ход сортировки пошагово:
[-6, -7, -10, -11, 0, -65, -32]
[-7, -6, -10, -11, 0, -65, -32]
[-10, -7, -6, -11, 0, -65, -32]
[-11, -10, -7, -6, 0, -65, -32]
[-11, -10, -7, -6, 0, -65, -32]
[-65, -11, -10, -7, -6, 0, -32]
отсортированный список:
[-65, -32, -11, -10, -7, -6, 0]
```

```
Сортировка 'вставками:'
первоначально: [5, 1, 4, 2, 8]
ход сортировки пошагово:
[5, 1, 4, 2, 8]
[1, 5, 4, 2, 8]
[1, 4, 5, 2, 8]
[1, 2, 4, 5, 8]
отсортированный список:
[1, 2, 4, 5, 8]
```

Рисунок 6 – Два примера сортировки вставками

5. Сортировка Шелла.

Сортировка называется по имени автора Дональда Шелла, и также потому, что действие этого метода часто иллюстрируется рядами морских раковин, перекрывающих друг друга (по-английски «shell» - «раковина»). Общая идея заимствована из сортировки вставками и основывается на уменьшении шагов. Сортируются все элементы, отстоящие друг от друга на d позиций, d уменьшается, процедура повторяется, в конце сортируются все соседние элементы.

Среднее время работы алгоритма зависит от длин промежутков d, на которых будут находиться сортируемые элементы исходного массива на каждом шаге, d можно выбрать, например, так: $d_1=n/2$, …, $d_i=d_{i-1}/2$, $d_k=1$.

Объяснение на примере: дан массив «32 95 16 82 24 40 66 35 19 75 54 40 43 93 68», надо отсортировать его по возрастанию:

Первый проход d=5 (сортируются каждые пятые элементы между собой, для визуального удобства используются одинаковые цвета):

(32 95 16 82 24 66 35 19 75 54 40 43 93 68) →
(32 95 16 82 24 66 35 19 75 54 40 43 93 68) →
(32 95 16 82 24 40 35 19 75 54 66 43 93 68) →
(32 35 16 82 24 40 43 19 75 54 66 95 93 68) →
(32 35 16 82 24 40 43 19 75 54 66 95 93 68) →
(32 35 16 68 24 40 43 19 75 54 66 95 93 82) →
(32 35 16 68 24 40 43 19 75 54 66 95 93 82)

Второй проход d=3 (сортируются каждые три элемента между собой):

(32 35 16 68 24 40 43 19 75 54 66 95 93 82) →
(32 35 16 68 24 40 43 19 75 54 66 95 93 82) →
(32 35 16 43 24 40 54 19 75 68 66 95 93 82) →
(32 19 16 43 24 40 54 35 75 68 66 95 93 82) →
(32 19 16 43 24 40 54 35 75 68 66 95 93 82) →

Третий проход d=1:
(32 19 16 43 24 40 54 35 75 68 66 95 93 82) →
(16 19 24 32 35 40 43 54 66 68 75 82 93 95)

Конкретная последовательность шагов может быть и другой, но последний шаг равен 1. Следует избегать последовательностей, которые являются степенями числа 2 - по математически сложным соображениям они уменьшают эффективность сортировки (но сортировка по-прежнему работает).

6. Быстрая сортировка.

Быстрая сортировка, часто называемая qsort, придуманная Хоаром обычно считается лучшим из существующих в настоящее время алгоритмом сортировки общего назначения. Быстрая сортировка построена на идее деления по спискам. Общая процедура: выбрать некоторое значение, называемое компарандом, и разбить массив на части - все элементы, меньшие компаранда - перемещаются влево, равные – в середину, а большие - направо. Потом этот процесс повторяется для каждой части (левой, правой) до тех пор, пока массив не будет отсортирован. Значение компаранда можно выбрать произвольно, далее в качестве компаранда будет выбираться первый элемент списка. Поясню на примере: отсортировать по возрастанию «-5 6 4 0 -45 34 8 -5 1 3 -2 -56».

Выбираем компаранд (=первый элемент) и разбиваем список на 3 части:
(-5 6 4 0 -45 34 8 -5 1 3 -2 -56) смотрим на массив меньших компаранда →
(-45 -56) сортируем, запоминаем, получаем (-56 -45), справа дописываем элементы, равные компаранду (их два), получаем и запоминаем (-56 -45 -5 -5)*, смотрим правую часть и повторяем рекурсивно:
(6 4 0 34 8 1 3 -2) за компаранд берем первый элемент списка, смотрим на массив меньших компаранда=6, то есть слева (а справа остается 34 8) →
(4 0 1 3 -2) сортируем рекурсионно, компаранд=4 →
(0 1 3 -2) сортируем рекурсионно (справа осталось 4 и 34 8), компаранд=0 →
(1 3) сортируем: получился отсортированный список (–2 0 1 3 4 6)**,
справа оставались (34 8), сортируем: (8 34)*** →
теперь соединяя все списки, получаем (*+**+***):
(-56 -45 -5 -5 -2 0 1 3 4 6 8 34).

На рисунке 7 отражен ход рассуждений быстрой сортировки.

```
'Быстрая' сортировка Хоара:
первоначальный массив:
[-5, 6, 4, 0, -45, 34, 8, -5, 1, 3, -2, -56]
ход сортировки пошагово:
[-5, 6, 4, 0, -45, 34, 8, -5, 1, 3, -2, -56]
[-45, -56]
[6, 4, 0, 34, 8, 1, 3, -2]
[4, 0, 1, 3, -2]
[0, 1, 3, -2]
[1, 3]
[34, 8]
[-56, -45, -5, -5, -2, 0, 1, 3, 4, 6, 8, 34]
отсортированный список:
[-56, -45, -5, -5, -2, 0, 1, 3, 4, 6, 8, 34]
```

```
'Быстрая' сортировка Хоара:
первоначальный массив:
[8, 7, 2, 26, 6, 33, -3, 30]
ход сортировки пошагово:
[8, 7, 2, 26, 6, 33, -3, 30]
[7, 2, 6, -3]
[2, 6, -3]
[26, 33, 30]
[33, 30]
[-3, 2, 6, 7, 8, 26, 30, 33]
отсортированный список:
[-3, 2, 6, 7, 8, 26, 30, 33]
```

Рисунок 7 – Ход рассуждений при использовании быстрой сортировки

3 Расчетно-графическая работа №3. Работа с фреймворками Python.

Цель работы: изучение основ работы с фреймворками.

3.1 Рабочее задание

В группе необходимо разбиться на небольшие команды, и в каждой команде все студенты равнозначно, помогая друг другу, согласованно, должны освоить одну из тем (как результат освоения представить реализованный рабочий проект, например, сайт или игру, написанную или адаптированную с помощью выбранного фреймворка), а также предоставить отчеты по одному из фреймворков:

1) Django.
2) Flaskr.
3) Kivi.
4) pyGame.
5) своя тема по согласованию с преподавателем.

3.2 Методические указания к выполнению работы

В процессе работы над командным проектом, всем членам команды надо установить необходимое программное обеспечение и показать работу на ноутбуке или персональном компьютере (вариант - видеоотчета).

В конце работы индивидуально каждым из студентов должен быть представлен отчет, включающий в себя описание хода работ (текстовое и принскрины) и собственное видение и выводы сравнения работы выбранного фреймворка (листинг, его обязательное построчное объяснение, общие критерии оценки работы и адаптации кода на других вычислительных машинах, возникшие сложности и способы их решения).

Список литературы

1 Стандарт организации учебно-методические и учебные работы СТ НАО 56023-1910-04-2014.

2 Язык программирования Python. Сузи Р.А. Учебное пособие. - М.: Интернет Университет информационных технологий, 2007. – 327 с.

3 Марк Лутц. Программирование на Python. Тома 1 и 2, 4-е издание. – Пер. с англ. - СПб.: Символ-Плюс, 2011. - 992 с.

4 Саммерфилд М. Программирование на Python 3. Подробное руководство. Пер. с англ. Киселев А. – М.: Символ-Плюс, 2009. - 608 с.

5 Доусон М. Программируем на Python. - СПб.: Питер, 2014. - 416 с.

6 http://pythonworld.ru/

7 Видеолекции на Youtube (открытая библиотека видеолекций): https://www.youtube.com/watch?v=xhoX3-NdM9k

Сводный план 2015 г., поз.150

Зуева Екатерина Александровна

ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ

Методические указания по выполнению расчетно-графических работ
для студентов специальности
5В100200 – Системы информационной безопасности

Редактор Л.Т. Сластихина
Специалист по стандартизации Н.К. Молдабекова

Подписано в печать __ __ _____ Формат 60х84 1/16.
Тираж 30 экз. Бумага типографская №1
Объем 1.9 уч.-изд.л. Заказ № __ Цена 950 тенге

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи»
050013, Алматы, ул. Байтурсынова, 126