

Федеральное агентство связи

**Федеральное государственное образовательное бюджетное учреждение
высшего профессионального образования**

**ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ**

**ЭЛЕКТРОННАЯ
БИБЛИОТЕЧНАЯ СИСТЕМА**

Самара

Государственное бюджетное образовательное учреждение высшего
профессионального образования
«Поволжский государственный университет телекоммуникаций
и информатики»

Методические указания на проведение лабораторной работы
«Исследование QR-кода»

по дисциплине «Информатика»,
(специальности 090302, 200700, 210400)



Автор-составитель
доц., к.т.н. **Алексеев А.П.**
Под общей редакцией Алексеева А.П.

Рецензент д.т.н., проф. Карташевский В.Г.

Самара, 2013 г.

Введение

Современные телекоммуникационные технологии стремительно внедряются в нашу повседневную жизнь. Удивительные возможности помехоустойчивого кодирования можно рассмотреть на примере QR-кодов.

Информация кодируется при помощи черных и белых квадратов, которые после сканирования превращаются в логические единицы и нули.

Формирование изображений матриц ведется таким образом, чтобы обеспечить наиболее благоприятные условия для работы сканирующей аппаратуры. Для этого в процессе кодирования создается восемь матриц и выбирается лучшая из них. Появляющиеся в процессе эксплуатации искажения рисунка матрицы устраняются путем использования помехоустойчивого кода Рида-Соломона.

Умение работать с QR-матрицами и понимать принципы кодирования и декодирования должно стать нормой для современного образованного человека. В процессе выполнения работы студенты учатся использовать современные телекоммуникационные средства: мобильные телефоны, смартфоны, планшеты.

Данная работа является ступенькой для плавного перехода от кодирования к шифрованию (криптографии).

Отчет по лабораторной работе должен содержать скриншоты (фотографии) матриц с необходимыми комментариями. Приветствуются самостоятельные исследования по вопросам, не затронутым в методических указаниях.

Автор выражает благодарность старшему преподавателю ПГУТИ Макарову М.И., обратившему внимание на интересные возможности QR-кода.

Лабораторная работа «Исследование QR-кода»

1. Подготовка к работе

По указанной литературе изучить назначение QR-кодов, способы кодирования и декодирования, ответить на контрольные вопросы.

2. Контрольные вопросы

- 2.1. Для каких целей используются QR-коды?
- 2.2. Какие специализированные сайты можно использовать для кодирования и декодирования QR-кодов?
- 2.3. В каком году изобретен QR-код?
- 2.4. В каком графическом формате целесообразно сохранять QR-коды (PNG, JPEG, BMP, SVG)?
- 2.5. Что такое уровень коррекции ошибок?
- 2.6. Сколько уровней коррекции ошибок можно сформировать в QR-кодах?
- 2.7. Какой алгоритм используют в QR-кодах для повышения их помехоустойчивости?
- 2.10. В чем преимущество 2D-кодов по сравнению со штрих - кодами?
- 2.11. Могут ли мобильные устройства считывать цветные QR-коды?
- 2.12. Как вручную определить уровень коррекции ошибок по изображению матрицы QR-кода?
- 2.13. Как по матрице определить версию QR-кода?
- 2.14. Сколько версий QR-кодов существует?
- 2.15. Как с помощью матрицы определить, какой вид маски использован при кодировании?
- 2.16. С какой целью при кодировании апробируется восемь видов различных масок?
- 2.17. Для каких областей матрицы используется статическая (неизменная) маска?
- 2.18. По заданной преподавателем формуле определите конфигурацию маски.
- 2.19. Каков алгоритм вычисления штрафных баллов?

3. Задания на выполнение лабораторной работы

3.1. Задание 1. Аппаратно-программное декодирование QR-кода

С помощью фотокамеры мобильного телефона (смартфона, нетбука, планшета) или специализированных Web-сайтов декодировать коды, приведенные в табл.3.1.

3.2. Задание 2. Формирование QR-кода (создание визитки)

Создать собственную визитку. Для этого с помощью генератора QR-кода закодировать свою фамилию, имя, группу и полное название ВУЗа.

3.3. Задание 3. Экспериментальное исследование помехоустойчивости QR-кода

Используя QR-код, полученный в предыдущем задании, выполнить следующие действия.

1. Изменить ориентацию изображения QR-кода (или фотокамеры). Произвести считывание (декодирование) информации.
2. Изменить масштаб изображения матрицы QR-кода. Произвести считывание информации. Определить минимально допустимые размеры картинки, при которых считывание информации еще возможно.
3. Нанести на изображение визитки 5 черных точек в произвольных местах матрицы. Попытаться считать информацию.
4. Дополнительно нанести на изображение визитки 5 белых точек. Произвести считывание информации.
5. Дополнительно к п.4 на изображении визитки в произвольном месте провести черную линию. Сделать попытку считывания информации.
6. Дополнительно к п.5 на изображении визитки провести белую линию. Произвести считывание информации.

3.4. Задание 4. Определение версии QR-кода

Определить версию кода для матрицы из таблицы 3.2.

3.5. Задание 5. Определение уровня коррекции ошибок

Определить уровни коррекции ошибок для двух матриц (таблицы 3.1 и 3.2).

3.6. Задание 6. Определение вида использованной маски

Определить вид использованных масок для двух матриц (таблицы 3.1 и 3.2).

3.7. Задание 7. Определение формата представления данных

Определить формат представления данных для двух матриц (таблицы 3.1 и 3.2).

3.8. Задание 8. Ручное декодирование QR-кода

Вручную декодировать сообщение, приведенное в табл. 3.3.

В представленных матрицах верхние пять строк умышленно срезаны (для исключения возможности аппаратно-программного декодирования).

В отчете следует подробно описать все этапы ручного декодирования и проиллюстрировать выполненные операции с помощью рисунков. Итогом выполнения задания должно стать сообщение, записанное с учетом верхнего и нижнего регистра (заглавные и строчные буквы).









3.9. Задание 9. Вычисление штрафных баллов

Произвести расчет штрафных баллов по правилу начисления штрафа за каждую группу из пяти или более одноцветных пикселей в одной строке (или столбце).

Для расчета использовать матрицы, приведенные в табл. 3.1.

ЭБС ПШУТМ

Таблица 3.1

Вар.	QR-код	Вар.	QR-код
1		2	
3		4	
5		6	
7		8	

Продолжение таблицы 3.1















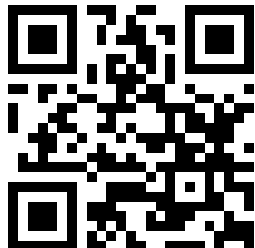

Вар.	QR-код	Вар.	QR-код
9		10	
11		12	
13		14	
15		16	

Таблица 3.2

Вар.	QR-код	Вар.	QR-код
1		2	
3		4	
5		6	
7		8	

Продолжение таблицы 3.2

























Вар.	QR-код	Вар.	QR-код
9		10	
11		12	
13		14	
15		16	

Таблица 3.3

Вар.	QR-код	Вар.	QR-код
1		2	
3		4	
5		6	
7		8	

Продолжение таблицы 3.3

Вар.	QR-код	Вар.	QR-код
9		10	
11		12	
13		14	
15		16	

4. Методические указания

QR-код (англ. quick response — быстрый отклик) — матричный код, разработанный японской компанией «Denso-Wave» в 1994 году [1].

Очевидно, что вместимость одной кодовой матрицы не является безграничной. Максимальное количество символов, которые можно поместить в одну матрицу, зависит от вида кодируемой информации (цифры, буквы, двоичный код, иероглифы), уровня коррекции ошибок и версии используемого QR-кода. Для версии 40 в одной матрице при минимальном уровне коррекции ошибок можно разместить 7089 десятичных цифр, либо 4296 букв и цифр, либо 2953 двоичных данных, либо 1817 иероглифов.

Наибольшее распространение рассматриваемый код получил среди абонентов мобильной связи. Пользователь может моментально занести в свой мобильный телефон (нетбук) текстовую информацию, контакты в адресную книгу, перейти по web-ссылке, отправить SMS-сообщение и т. д.

С помощью QR-кодов удобно делать визитки и размещать их на майках, фуражках, значках. Коды открывают большие возможности для рекламы. За счет размещения QR-кода на рекламном щите, плакате (постере) или объявлении можно привлечь большое число клиентов.

Для того чтобы сделать туристические маршруты более информативными, на улицах г. Львова (Украина) размещено более 80 изображений QR-кодов. Ими оснащены коммерческие и культурные объекты, в том числе памятники, архитектурные сооружения.

QR-коды используются для маркировки ответственных деталей. Гравировка металлической поверхности позволяет наносить код на детали, подвергающиеся воздействиям высоких температур, давлений или агрессивных химических веществ. Данный код имеет высокую помехоустойчивость. Даже при наличии повреждений (искажений) изображения на площади 30% всё еще есть возможность безошибочно считать информацию.

С помощью QR-кодов можно получать прогнозы времени прибытия транспорта на конкретную остановку. Например, транспортный оператор г. Самары планирует разместить наклейки с QR-кодами на остановочных павильонах. Можно считать QR-код с названием остановки фотокамерой своего мобильного телефона. Это позволяет сразу открыть страницу с прогнозами прибытия транспорта на данную остановку.

QR-код не является единственным вариантом двумерного штрих-кода. Известны и другие форматы: DataMatrix, ScanLife EZcode и Microsoft Tag (Tag).

Три квадрата в углах изображения QR-кода позволяют устройству правильно определять его пространственную ориентацию. Черные и белые точки изображения преобразуются считывающим устройством в двоичные числа. Затем производится обработка цифровой информации. На матрице размещается системная информация, информационные биты, корректирующие биты [2]. При обработке считанного изображения используется алгоритм Рида-Соломона, который позволяет устранить треть возможных искажений матрицы.



Рис. 4.1. Разновидности двумерных кодов

QR-коды можно создавать с различной степенью помехозащищенности. Чем выше помехозащищенность, тем больше избыточной информации содержит матрица и тем меньше остается места для информационных битов.

Матрицу можно создать с одним из четырех уровней коррекции ошибок. Эти уровни определяют долю информации, подлежащую восстановлению за счет использования корректирующего кода.

Таблица 4.1

Уровень коррекции ошибок	Код	Допустимые искажения
L	01	7%
M	00	15%
Q	11	25%
H	10	30%

Уровни L и M рекомендуется использовать для печати кодов на пригласительных билетах, объявлениях, визитках, плакатах, рекламных щитах и т.д. Уровни Q и H применяют для маркировки промышленных деталей, когда существует высокая вероятность повреждения изображения кода.

Таблица 4.2 показывает, как выглядит код слова «Проба» при различных уровнях коррекции ошибок. Визуально можно заметить, что с увеличением уровня помехоустойчивости число элементов в матрице растет.

Таблица 4.3 демонстрирует высокую степень помехоустойчивости QR-кодов. Даже на перечеркнутых матрицах есть возможность правильно декодировать информацию.

Следует обратить внимание, что считывание информации прекращается в случаях, когда существенно искажен один из трех прямоугольников, отвечающих за определение ориентации матрицы (см. табл. 4.4). В то же время наличие двадцати цветных точек на изображении матрицы не приводит к снижению надежности считывания информации (см. табл. 4.5).

Таблица 4.2





7%		15%	
25%		30%	

Таблица 4.3





1		2	
3		4	

Таблица 4.4



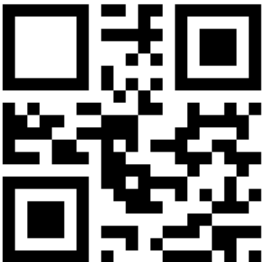
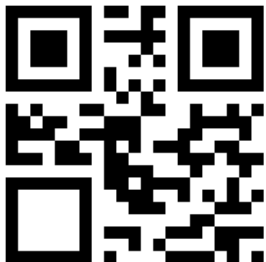
1		2	
3		4	

Таблица 4.5

1		2	
3		4	

Структура матрицы

Информационные биты, контрольные биты и системная информация располагаются в определенных местах матрицы. Порядок размещения всех элементов матрицы определены правилами, которые описаны в спецификации [4].

На каждой матрице имеется три указателя ориентации 1 (УО). Между тремя УО проходят две линии синхронизации 2, которые состоят из чередующихся белых и черных модулей (пикселей). Вблизи левого нижнего указателя ориентации размещается черный пиксель 3 (см. рисунок 4.2).

В настоящее время выпущено 40 версий QR-кодов. На матрицах QR-кодов, начиная с версии 2, имеются метки центровки 4. В версии 1 такой метки нет, но в старших версиях QR-кода можно насчитать 46 таких меток.

В областях 5 для версий старше шестой размещается информация о версии кода. В четырех модулях 6 содержится код, которые определяет, какой вид данных содержится в матрице. Системная информация размещается в областях 7 и 8.

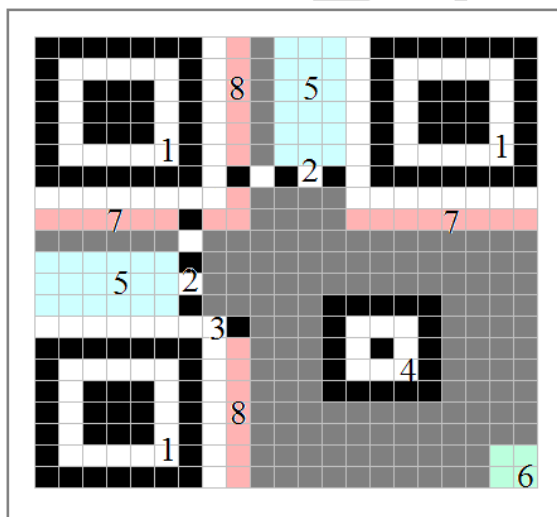


Рис.4.2. Структура матрицы

Информация об уровне коррекции ошибок и виде использованной маски располагается в системной строке (см. рис. 4.3). Разряды 14 и 13 содержат сведения об используемом уровне коррекции ошибок. Разряды 12, 11 и 10 показывают, какая маска использована для наложения на информационные и корректирующие биты. Три бита позволяют сформировать коды для восьми разных масок.

Системная информация дублируется. Один раз 15 системных битов размещают в столбце 8 (счет элементов ведется, начиная с 0 из левого верхнего угла матрицы).

Второй раз системную информацию размещают в строке 8.

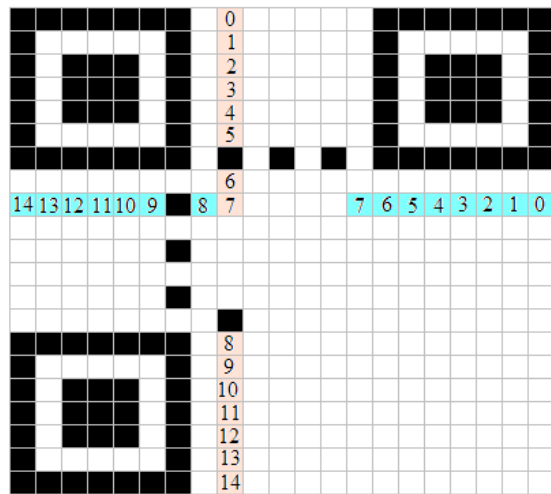


Рис. 4.3. Дублирование системной информации

ЭБС ПШУТИИ

4.1. Методические указания к заданию 3.1

Наиболее просто операция декодирования выполняется с помощью мобильного телефона со встроенной фотокамерой и установленной программой считывания. На ряде моделей мобильных телефонов нет программ для распознавания QR-кодов. В этом случае рекомендуются установить следующие программы:

для Android: Google Goggles, QuickMark, Barcode Scanner, Barcode2file, QR Droid, NeoReader, ixMAT Scanner, 2D-код, Elinext UPC, I-Nigma, QR Droid;

для iPhone и iPad: RedLaser, QR Reader for iPhone, Bakodo - Barcode Scanner and QR Bar Code Reader, AT&T Code Scanner, Elinext UPC, Ценометр;

для Windows Phone 7: приложение уже встроено в поиск (Обновление Mango);

для Symbian OS: QuickMark, Kaywa reader, Nokia barcode reader, I-Nigma, UpCode, NeoReader, BeeTag;

для Java: Kaywa reader, I-Nigma, UpCode;

для Bada: BeeTagg, Quick QR Reader;

для Maemo: mbarcode;

для Windows Mobile: QuickMark, I-Nigma;

для BlackBerry OS: приложение уже встроено в App World.

При отсутствии у пользователя устройств распознавания QR-кода кодирование и декодирование можно выполнить с помощью скриптов, к которым есть свободный доступ в Интернет.

В качестве специализированных Web-сайтов (генераторов и распознавателей QR-кодов), предназначенных для кодирования и декодирования QR-кодов, можно использовать:

- <http://qrcoder.ru>
- <http://qr-coder.net/>
- <http://qr.foxtools.ru>
- <http://qrrussia.ru/>
- <http://www.qurify.com/ru/>
- <http://creambee.ru/>
- <http://www.qrstuff.com/>
- <http://www.qrcodegenerator.ru/>

Допустимо для кодирования и декодирования использовать любой другой сайт, найденный самостоятельно в Интернет.

В отчете нужно указать, какое аппаратное средство использовано или с помощью какого сайта произведено кодирование (декодирование) информации.

4.2. Методические указания к заданию 3.2

Если формирование кода ведется с помощью аппаратного средства, то нужно ознакомиться с инструкцией по работе с мобильным телефоном, нетбуком или планшетником. Полученное изображение кода можно демонстрировать преподавателю непосредственно на мобильном телефоне (нетбуке, планшетнике). Допустимо результат кодирования представлять в виде графического файла. Наилучшие графические форматы для подобных изображений PNG или GIF (в этих случаях размеры файлов будут наименьшими).

Если кодирование ведется с помощью специализированного сайта, то необходимо сделать фотографию (скриншот) полученного кода. Преподавателю следует представить графический файл либо фотографию кода, сделанную цифровым фотоаппаратом. Менее рациональный вариант - представить матрицу в распечатанном на принтере виде. При наличии у студента большого количества свободного времени матрицу можно аккуратно перерисовать в отчет карандашом ☺.

4.3. Методические указания к заданию 3.3

При выполнении этого задания целесообразно использовать растровый графический редактор типа MS Paint. С его помощью удобно на изображение матрицы наносить точки и проводить линии.

Результаты сканирования с экрана монитора во многом зависят от качества дисплея и его настройки, а также от матрицы фотоаппарата мобильного телефона. Поэтому в отчете нужно указать тип использованного сканирующего устройства и дисплея.

Изменить масштаб рисунка можно, вставив его в текстовый редактор. Результаты исследований в этом задании нужно представить преподавателю в виде картинок (скриншотов) с комментариями. Примерный вид комментариев должен быть таким: «Считывание информации прошло успешно» или «Считать информацию не удалось».

При определении минимально допустимого размера матрицы длину сторон квадрата нужно указать в сантиметрах.

4.4. Методические указания к заданию 3.4

Существует 40 версий QR-кода. С увеличением номера версии кода объем помещаемых в матрицу данных увеличивается. Определить, к какой версии относится данная матрица, можно тремя способами:

- подсчитать, какое число пикселей содержит матрица;
- определить координаты меток центровки;
- для версий старше шестой существуют области на матрице, где содержится информация об используемой версии.

В таблице приведена информация о числе модулей (пикселей) для различных версий QR-кодов.

Табл. 4.4.1

Версия	Число модулей
1	21x21
2	25x25
3	29x29
4	33x33
...	...
40	177x177

Анализируя, приведенную таблицу, легко заметить линейную закономерность: очередная версия QR-кода отличается от предыдущей версии тем, что сторона матрицы увеличена на 4 модуля. Подсчитав число модулей, можно определить версию QR-кода для данной матрицы.

В таблице приведены сведения о координатах меток центровки для версий 2...13. Полная версия этой таблице приведена в Приложении 3.

Табл. 4.4.2

Версия	Строки		
2	6	18	
3	6	22	
4	6	26	
5	6	30	
6	6	34	
7	6	22	38
8	6	24	42
9	6	26	46
10	6	28	50
11	6	30	54
12	6	32	58
13	6	34	62

На следующем рисунке для примера показано изображение матрицы QR-кода версии 7.

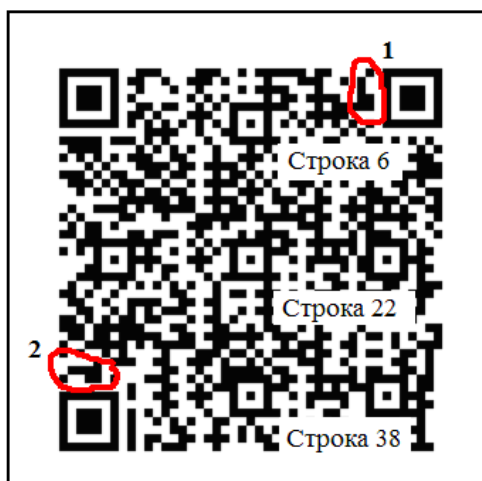


Рис. 4.4.1 Матрица QR-кода версии 7

Из рисунка видно, что шесть меток центровки расположены на пересечении строк и столбцов, определенных в таблице. Координаты определяются по черному модулю (пикселю), расположенному внутри белого квадрата. Метки центровки не ставятся в местах, где расположены указатели ориентации (три угла).

На матрицах, начиная с версии 7, имеются две области, где содержится закодированная информация об использованной версии. Эти области отмечены на рис. 4.4.1 цифрами 1 и 2. В табл. 4.4.3 приведены коды некоторых версий QR-кода. Полная версия таблицы приведена в Приложении 4.

Табл. 4.4.3

Версия	Код
7	001010010011111000
8	000111101101000100
9	100110010101100100
10	011001011001010100
11	01101111101110100
12	001000110111001100
13	111000100001101100

Таблицы 4.4.4 и 4.4.5 показывают, в каком порядке размещаются биты восемнадцатиразрядного кода в отмеченных областях.

Табл. 4.4.4

0	3	6	9	12	15
1	4	7	10	13	16
2	5	8	11	14	17

Табл. 4.4.5

0	1	2
3	4	5
6	7	8
9	10	11
12	13	14
15	16	17

4.5. Методические указания к заданию 3.5

Определить уровень коррекции ошибок в конкретном случае можно вручную. Так как системная информация дублируется, то сведения о имеющейся корректирующей способности кода можно отыскать в двух местах матрицы. Один раз 15 бит системной информации помещаются в строке 1 (отсчет слева направо), а второй раз - в столбце 2 (отсчет снизу вверх). Естественно, что эти два значения одинаковые. Сведения об уровне коррекции ошибок содержатся в первых двух битах системной информации. Эти два бита расположены в 14 и 13 разрядах системной строки (см. рис. 4.4).

На рисунке 4.5.1 первые два бита равны 1. Напомним, что логические единицы изображены в виде черных пикселей (модулей), а логические нули – в виде белых. Чтобы определить уровень коррекции нужно эти два бита сложить по правилу Исключающее Или с двоичной маской 10.

В результате сложения получим:

$$\begin{array}{r} 1\ 1 \\ \oplus 1\ 0 \\ \hline 0\ 1 \end{array}$$

Код 01 (см. табл. 4.1) соответствует уровню коррекции ошибок L (7%).

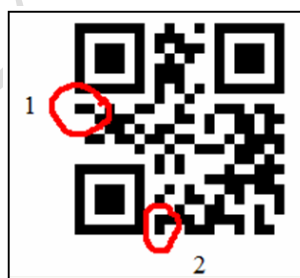


Рис. 4.5.1. Уровень коррекции ошибок 01

На следующем рисунке показана матрица, в которой использован иной уровень коррекции ошибок. Первые два бита во втором случае равны 10. Суммирование с маской 10 дает результат 00. Это говорит о том, что в этой матрице использован код с уровнем коррекции ошибок M (15%).

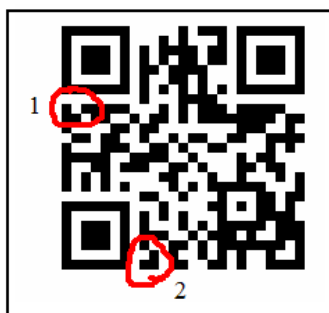


Рис. 4.5.1. Уровень коррекции ошибок 00

Аналогично производятся расчеты для матриц с уровнями коррекции ошибок Q и H. На этих матрицах системная информация начинается битами 01 и 00 соответственно. Суммирование с маской 10 даст коды 11 и 10 соответственно.

4.6. Методические указания к заданию 3.6

Для повышения надежности сканирования матрицы подвергают дополнительной обработке. Целью этой обработки является формирование такого рисунка матрицы, на котором было бы как можно меньше смежных модулей одинакового цвета. Простыми словами это требование можно сформулировать так: «Матрица должна быть пёстрой и на ней должно быть минимум однотонных сплошных протяженных участков» (масло в каше должно быть хорошо перемешано).

Цель достигается следующим образом: формируют 8 матриц с одинаковым информационным содержанием, но разного вида (разной формы). Для этого используют 8 масок, которые суммируют с исходной матрицей по правилу Исключающее ИЛИ. Полученные матрицы поочередно подвергают математической обработке, в результате которой выбирают одну матрицу. Отбор лучшей матрицы осуществляют путем вычисления штрафных баллов. Для оптимальной матрицы число штрафных баллов будет минимальным.

Восемь масок кодируются трехразрядными числами от 000 до 111. Математически маски описываются формулами, приведенными в табл. 4.6.1.

Табл. 4.6.1

Код маски	Соотношение
000	$(i + j) \bmod 2 \equiv 0$
001	$i \bmod 2 \equiv 0$
010	$j \bmod 3 \equiv 0$
011	$(i + j) \bmod 3 \equiv 0$
100	$((i \operatorname{div} 2) + (j \operatorname{div} 3)) \bmod 2 \equiv 0$
101	$(i \cdot j) \bmod 2 + (i \cdot j) \bmod 3 \equiv 0$
110	$((i \cdot j) \bmod 2 + (i \cdot j) \bmod 3) \bmod 2 \equiv 0$
111	$((i \cdot j) \bmod 3 + (i \cdot j) \bmod 2) \bmod 2 \equiv 0$

В таблице использованы обозначения: j – номер строки матрицы; i – номер столбца; mod – операция поиска остатка от целочисленного деления; div – операция деления.

Порядок определения координат модулей иллюстрирует рис.4.6.1.

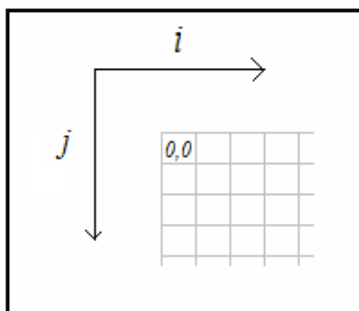


Рис. 4.6.1. Система координат

Формулы используются следующим образом: если координаты рассматриваемого модуля матрицы удовлетворяют данному соотношению, то этот модуль инвертируется. Рассчитанные по формулам таблицы 4.6.1 «узоры» приведены в Приложении 1.

В данном задании требуется лишь определить, какая из восьми масок применена для преобразования исследуемой матрицы. Чтобы выполнить задание, нужно считать три бита, расположенных в разрядах 12, 11 и 10 (см. рис. 4.4). Считанные три бита нужно сложить по правилу Исключающее ИЛИ с числом 101. Полученный результат определяет вид использованной маски в данной матрице (табл. 4.6.1).

4.7. Методические указания к заданию 3.7

При кодировании конкретных сообщений их содержимое бывает самым разнообразным. В одном случае это только десятичные цифры, в другом – буквы и цифры, в третьем – двоичные цифры, в четвертом – иероглифы.

В зависимости от вида используемых данных запись битов в матрицу происходит в разных форматах. Вид формата указывается с помощью четырех битов, расположенных в области 6 (рис. 4.2).

Следует обратить внимание, что в области 6 информация о формате данных «искажена» маской. Причем в каждом случае маска может быть любой из восьми допустимых. Поэтому для получения истинного значения кода формата данных нужно обязательно предварительно определить вид использованной маски (см. п.4.6).

Порядок нумерации битов показан на рисунке, причем старший разряд кода формата находится в клетке 1.

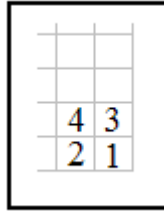


Рис. 4.7.1. Порядок считывания битов

Названия форматов при различных значениях кода указаны в табл. 4.7.1.

Табл. 4.7.1

Код	Формат
0000	Terminator (End of Message)
0001	Цифровой
0010	Символьный
0011	Structured Append
0100	Двоичный (байт, 8 бит)
0101	FNC1 (First position)
0111	ЕСІ
1000	Японский
1001	FNC1 (Second position)

4.8. Методические указания к заданию 3.8

Предположим, что требуется вручную декодировать матрицу QR-кода, показанную на рис. 4.8.1.

Вначале определим, какой вид маски использован при её кодировании. В разрядах 12, 11 и 10 системной строки (рис. 4.3) записано двоичное слово 111. Сложение со статической маской 101 дает двоичное число 010. По таблице 4.6.1 определяем формулу, использованную для формирования маски:

$$j \bmod 3 \equiv 0. \quad (4.6.1)$$

По формуле 4.6.1 определим вид расчетной маски (рис. 4.8.2). Рисунок 4.8.2 нужно трактовать следующим образом: если на маске имеется черный пиксель (модуль), то пиксель матрицы с такими же координатами нужно проинвертировать. У этого правила большое число исключений. На рис. 4.8.3 показаны серым цветом модули, которые инвертировать не следует.

Анализируя рисунок 4.8.3, можно заметить, что от изменения с помощью маски защищены области матрицы, где расположены указатели ориентации, линии синхронизации, системная информация (серые модули на рис. 4.8.3). Скорректированную таким образом расчетную маску назовём фактической маской.

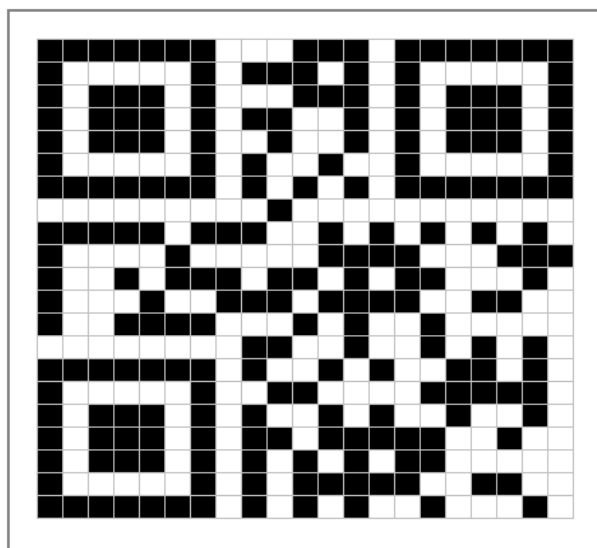


Рис. 4.8.1. Матрица

Процесс наложения фактической маски на матрицу (рис. 4.8.1) показан на рис. 4.8.4.

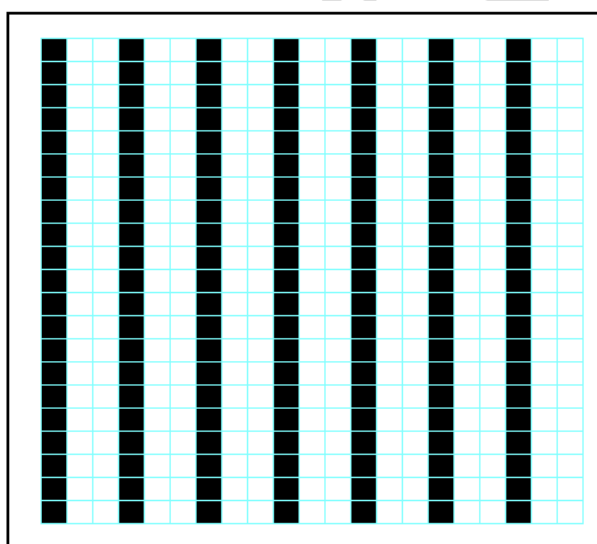


Рис. 4.8.2. Расчетная маска

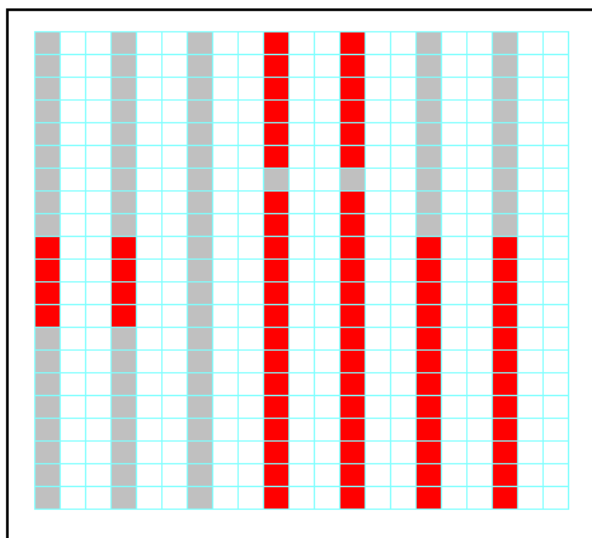


Рис. 4.8.3. Фактическая маска

Наибольший интерес при выполнении этого задания представляет область матрицы, расположенная в правом нижнем углу матрицы. Здесь находятся информационные биты. Рассмотрим детально эту зону (рис.4.8.5). Рисунок 4.8.5 а) показывает фрагмент исходной матрицы, который должен быть подвержен обработке. На рис. 4.8.5 б) выделены модули, которые должны быть проинвертированы. Следует обратить внимание, что верхняя граница рисунка совпадает с системной строкой. Результат взаимодействия маски и матрицы показан на рис. 4.8.5 в).

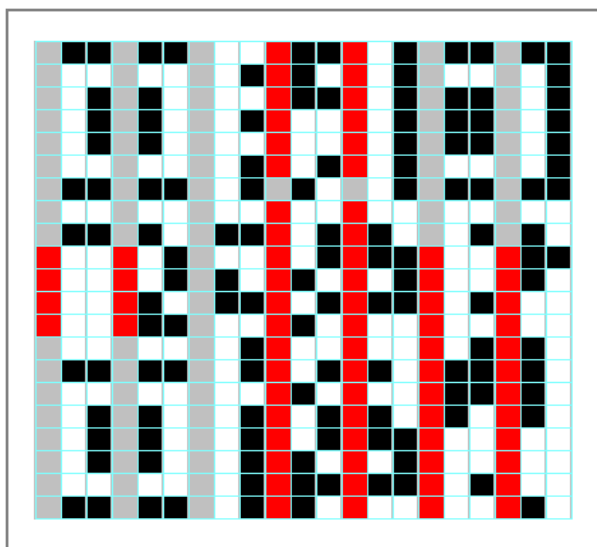


Рис. 4.8.4. Совмещение матрицы и фактической маски

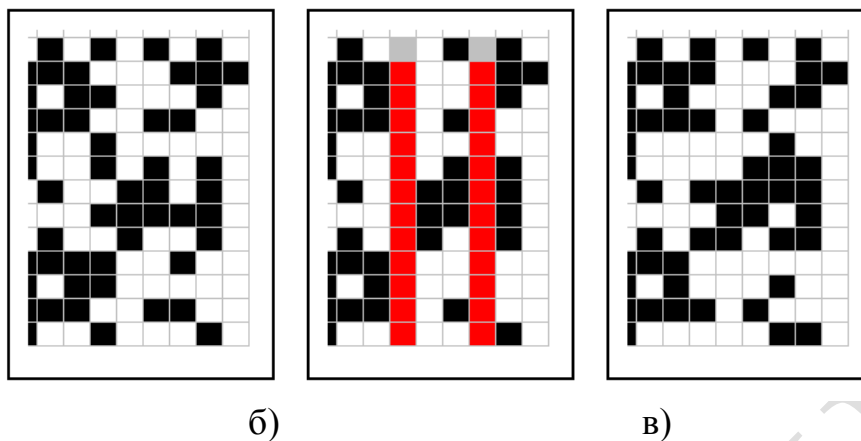


Рис. 4.8.5. Инвертирование выделенных модулей

Из рисунка 4.8.5 в) видно, что модули третьего и шестого столбца справа были проинвертированы.

Теперь есть возможность выполнить ручное декодирование матрицы 4.8.5 в). Четыре бита в клетках 1...4 дают код 0100, который говорит о том, что в данной матрице используется двоичный формат данных (см. табл.4.7.1).

Порядок считывания декодируемой информации показан на рис. 4.8.6.

В соответствии с двоичным форматом данных следующие 8 бит (клетки 5...12) указывают, сколько символов содержится в сообщении. Двоичное число 00000101 говорит о том, что данное сообщение содержит пять символов.

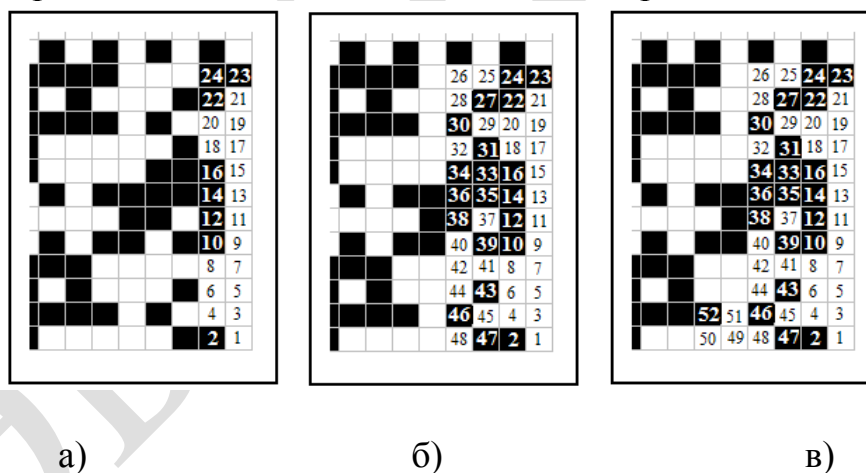


Рис. 4.8.6. Нумерация информационных модулей

В очередных восьми клетках 13...20 (см. рис. 4.8.6 а) размещено двоичное слово 01010000. Перевод этого байта в десятичную систему счисления дает число 80. По таблице CP-1251 (см. Приложение 2) определяем, что данному коду соответствует латинская буква «Р».

Следует обратить внимание на порядок нумерации модулей (рис. 4.8.6). Номера располагаются друг за другом в виде змейки.

Результаты декодирования сообщения приведены в таблице 4.8.1.

Табл. 4.8.1

Номера модулей	Биты	Десятичное число	Символ
13...20	01010000	80	P
21...28	01110000	114	r
29...36	01101111	111	o
37...44	01100010	98	b
45...52	01100001	97	a

Таким образом, в результате ручного декодирования было определено, что матрица (рис. 4.8.1) содержит слово «Proba».

4.9. Методические указания к заданию 3.9

При кодировании сообщения формируют восемь матриц с одинаковым содержанием, но разной формы. Для выбора одной из восьми конкурирующих матриц в стандарте [4] определены четыре правила расчета штрафных баллов.

Согласно первому правилу штраф назначается за каждую группу из пяти или более одноцветных пикселей в одной строке (или столбце). Второе правило дает штраф за каждый одноцветный квадрат размером 2x2 пикселя. Третье правило определяет штраф, если на матрице есть области, которые похожи на указатели определения ориентации. Четвертое правило начисляет штраф, если более половины пикселей одного цвета.

Целью расчетов штрафных баллов является выбор наиболее «пёстрой» матрицы из восьми возможных вариантов.

В данной лабораторной работе расчет штрафных баллов будет происходить только в соответствии с первым правилом.

Для примера используем матрицу, показанную на рис. 4.8.1. Вычислим штрафные баллы построчно.

Первая сверху строка матрицы содержит большое число смежных черных пикселей (две группы однотонных пикселей). Это говорит о том, что для первой строки будет начислено большое число штрафных баллов.



Рис. 4.9.1. Первая строка матрицы

Расчет ведется следующим образом. Если в строке имеется группа из пяти смежных одноцветных пикселей, то начисляется три штрафных балла. За каждый последующий пиксель, который примыкает к группе, начисляется дополнительно по одному баллу. Таким образом, первая группа черных пикселей дает вклад $3 + 1 + 1 = 5$ баллов. Вторая группа из семи черных пикселей также дает 5 штрафных баллов. В общей сложности первая строка дала 10 штрафных баллов. Полученное число поместим в правый крайний столбец матрицы (рис.4.9.4).

Вторая сверху строка матрицы показана на следующем рисунке.



Рис. 4.9.2. Вторая строка матрицы

В этой строке есть две группы по пять белых пикселей. Эти две группы в сумме дают 6 штрафных баллов.



Рис. 4.9.3. Третья строка матрицы

Третья строка матрицы не содержит протяженных групп и поэтому не добавляет штрафных баллов.

Аналогичная методика подсчета штрафных баллов используется и для столбцов. На следующем рисунке показаны штрафные баллы по всем строкам и столбцам матрицы.

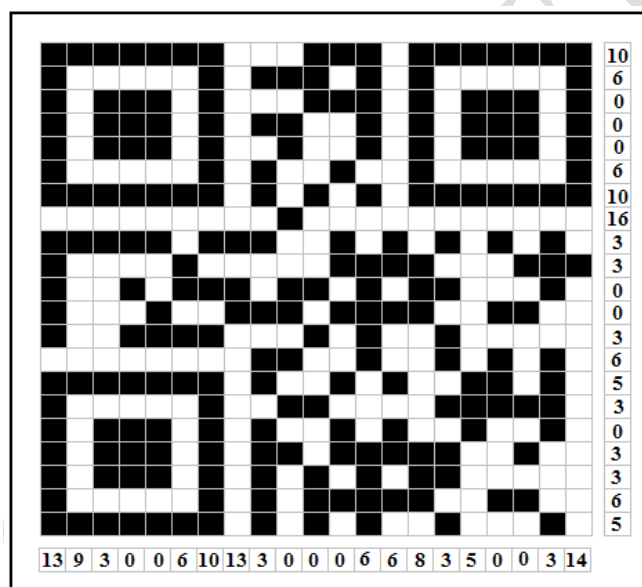


Рис. 4.9.4. Штрафные баллы

Затем рассчитываются суммарные значения штрафных баллов по строкам и по столбцам. Для рассматриваемого примера суммарное число штрафных баллов по строкам равно 88, а по столбцам 102. Общее число штрафных баллов, вычисленное по первому правилу для данной маски, дает число 190.

Используя все четыре правила расчета штрафных баллов, находят суммарную количественную оценку данной маски. Подобные оценки получают для всех восьми масок. Среди этих оценок выбирают минимальную оценку и для окончательного кодирования используют маску, которая имеет минимальное число штрафных баллов.

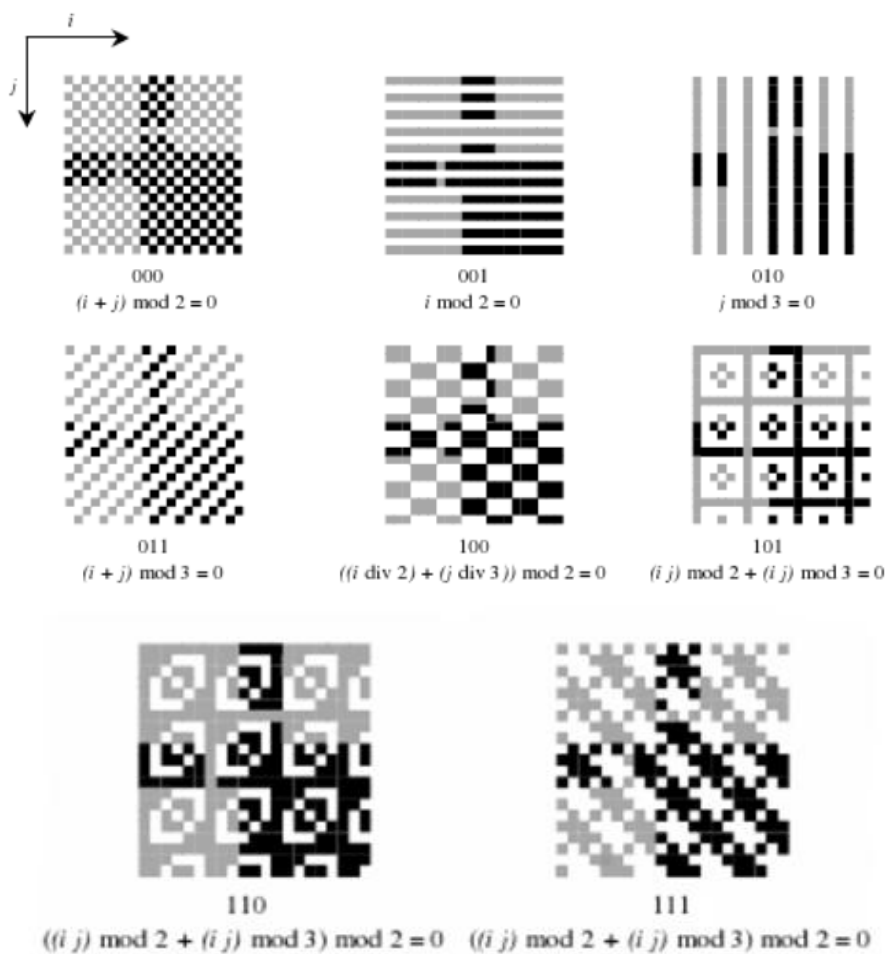
Примечание. В отчете необходимо привести матрицу, подобную показанной на рис. 4.9.4 (для своего варианта).

Список литературы

1. <http://ru.wikipedia.org/wiki/QR-%EA%EE%E4>
 2. <http://habrahabr.ru/post/127197/>
 3. http://kcoding.net/2012/qrcode_tutorials/
 4. INTERNATIONAL STANDART ISO/IEC 18004/ First edition 2000-06-
- 15.

ЭБС ПШУТИИ

Приложение 1. Фактические маски



Приложение 2. Таблица CP-1251

пробел	32	!	33	"	34	#	35	\$	36
%	37	&	38	'	39	(40)	41
*	42	+	43	,	44	-	45	.	46
/	47	0	48	1	49	2	50	3	51
4	52	5	53	6	54	7	55	8	56
9	57	:	58	;	59	<	60	=	61
>	62	?	63	@	64	A	65	B	66
C	67	D	68	E	69	F	70	G	71
H	72	I	73	J	74	K	75	L	76
M	77	N	78	O	79	P	80	Q	81
R	82	S	83	T	84	U	85	V	86
W	87	X	88	Y	89	Z	90	[91
\	92]	93	^	94	_	95	`	96
a	97	b	98	c	99	d	100	e	101
f	102	g	103	h	104	i	105	j	106
k	107	l	108	m	109	n	110	o	111
p	112	q	113	r	114	s	115	t	116
u	117	v	118	w	119	x	120	y	121
z	122	A	192	Б	193	В	194	Г	195
Д	196	Е	197	Ж	198	З	199	И	200
Й	201	К	202	Л	203	М	204	Н	205
О	206	П	207	Р	208	С	209	Т	210
У	211	Ф	212	Х	213	Ц	214	Ч	215
Ш	216	Щ	217	Ъ	218	Ы	219	Ь	220
Э	221	Ю	222	Я	223	а	224	б	225
в	226	г	227	д	228	е	229	ж	230
з	231	и	232	й	233	к	234	л	235
м	236	н	237	о	238	п	239	р	240
с	241	т	242	у	243	ф	244	х	245
ц	246	ч	247	ш	248	щ	249	ъ	250
ы	251	ь	252	э	253	ю	254	я	255

Приложение 3. Определение версии кода с помощью меток центровки

Версия	Строки, столбцы	Строки, столбцы	Строки, столбцы	Строки, столбцы	Строки, столбцы	Строки, столбцы	Строки, столбцы
2	6	18					
3	6	22					
4	6	26					
5	6	30					
6	6	34					
7	6	22	38				
8	6	24	42				
9	6	26	46				
10	6	28	50				
11	6	30	54				
12	6	32	58				
13	6	34	62				
14	6	26	46	66			
15	6	26	48	70			
16	6	26	50	74			
17	6	30	54	78			
18	6	30	56	82			
19	6	30	58	86			
20	6	34	62	90			
21	6	28	50	72	94		
22	6	26	50	74	98		
23	6	30	54	78	102		
24	6	28	54	80	106		
25	6	32	58	84	110		
26	6	30	58	86	114		
27	6	34	62	90	118		
28	6	26	50	74	98	122	
29	6	30	54	78	102	126	
30	6	26	52	78	104	130	
31	6	30	56	82	108	134	
32	6	34	60	86	112	138	
33	6	30	58	86	114	142	
34	6	34	62	90	118	146	
35	6	30	54	78	102	126	150
36	6	24	50	76	102	128	154
37	6	28	54	80	106	132	158
38	6	32	58	84	110	136	162
39	6	26	54	82	110	138	166
40	6	30	58	86	114	142	170

Приложение 4. Определение версии QR-кода с помощью системной информации

Версия	Двоичное слово
7	001010010011111000
8	000111101101000100
9	100110010101100100
10	011001011001010100
11	011011111101110100
12	001000110111001100
13	111000100001101100
14	010110000011011100
15	000101001001111100
16	000111101101000010
17	010111010001100010
18	111010000101010010
19	001001100101110010
20	011001011001001010
21	011000001011101010
22	100100110001011010
23	00011011111111010
24	001000110111000110
25	000100001111100110
26	110101011111010110
27	000001110001110110
28	010110000011001110
29	001111110011101110
30	101011101011011110
31	000000101001111110
32	101010111001000001
33	000001111011100001
34	010111010001010001
35	011111001111110001
36	110100001101001001
37	001110100001101001
38	001001100101011001
39	010000010101111001
40	100101100011000101