Ministry of Higher Education and Scientific Research
Relizane University
Faculty of Science and Technology
Mathematics Department

A Project Submitted of the Requirements for the
Master's Degree in Mathematics

Barcode and QR in Mathematics

Prepared by:

BACHIR FETHI

Jury Members:

President :	Mr Abdallah Beddani	(MCA)	Relizane University
Supervisor :	Mr Charef Beddani	(MCA)	Mostaganem University
Examiner :	Mr Habib Djourdem	(MCA)	Mostaganem University

2024/2025

Dedication

I wholeheartedly dedicate this work to those who have supported me throughout this journey:

To my dear parents – your love, sacrifices, and endless prayers are the foundation of everything I have achieved. I owe you more than words can ever express.

To my small but precious family – thank you for your patience, support, and for always being there through every challenge and success.

To my classmates and friends – your companionship, motivation, and shared moments made this journey enjoyable and meaningful.

To all of you – this work is a reflection of our shared efforts, and I am deeply grateful for every step we took together.

Acknowledgments

I would like to sincerely express my deepest appreciation to my supervisor, Dr. Charef Beddani, whose expert guidance, continuous encouragement, and insightful advice were invaluable throughout the development of this research.

My heartfelt thanks also go to the esteemed members of the committee Dr. Djourdem Habib and Dr. Beddani Abdallah, for their constructive comments, valuable suggestions, and support, which significantly enhanced the quality of this work.

I am grateful to my colleagues and friends for their encouragement and assistance. Lastly, I extend my profound gratitude to my family for their unwavering patience, understanding, and support throughout this journey.

Table of contents

Introduction

In today's digital era, data encoding systems such as **barcodes** and **QR codes** have become essential tools across a wide range of applications—from inventory management and logistics to digital payments, marketing, and identity verification. While these codes may appear simple at first glance, their construction and functionality are deeply rooted in sophisticated **mathematical concepts**, particularly in the fields of **number theory** and **algebra**.

Mathematics plays a crucial role in the reliable encoding, storage, and retrieval of data. Specifically, **modular arithmetic**, **matrix algebra**, **Diophantine equations**, and even aspects of **probability theory** and **error-correcting codes** form the backbone of how these encoding systems operate. These principles ensure not only data integrity and accuracy but also efficient and secure transmission of information.

In **one-dimensional barcodes** such as **UPC** (Universal Product Code) and **EAN** (European Article Number), **check digits** are computed using algorithms based on **modulo operations** to detect and prevent errors during scanning or manual data entry. On the other hand, **two-dimensional QR codes** involve more complex structures, where data is stored in a grid and protected using **error-correction codes** such as **Reed–Solomon codes**, which rely on **finite field algebra** and **linear algebraic techniques**.

This project aims to:

- Analyze the **mathematical foundations** that govern the structure and functionality of barcodes and QR codes.
- Demonstrate how **equations, number theory, and algebraic structures** are used in encoding and error detection.
- Explore the **practical implementation** of these mathematical ideas in real-world data systems.
- Highlight the significance of **algebra and number theory** in constructing robust, efficient, and scalable data encoding solutions.

Understanding the mathematical logic behind barcode and QR code systems not only enhances our appreciation for their design but also opens the door to developing improved encoding methods. This project ultimately reveals how **pure mathematical theories**, once regarded as abstract and theoretical, now power critical components of modern information systems.

Chapter

1

Barcodes and Data Encoding Systems

Linguistically, the word consists of two parts in English: **Bar**, which means column, and **Code**, which means data encoding. Together, they refer to encoding or encrypting a set of data into codes using a device. These codes are printed as dark columns on a white background in various shapes and sizes.

Barcode is a code representing certain data that can be read by computers. Linear barcodes may appear as lines, dots, hexagonal shapes, or other geometric patterns within images. Some are called two-dimensional codes. Although two-dimensional systems use more symbols than linear lines (threads), they are generally referred to as linear codes. Barcodes are designed to be read accurately and quickly by a device called a **Barcode Reader**.

Figure 1.1: Illustrative representation of a barcode

1.1 Barcode Components

A barcode consists of 13 digits:

- The first digit is specific to the company's system that printed the barcode.
- The next three digits represent the country code where the company is registered.
- The following three digits are specific to the company that owns the product.
- The last six digits represent the product's unique code.

1.2 Types of Barcodes

1.2.1 Numeric Encoding

There are 12 types:

1. **EAN-13**: European Article Numbering for retail trade.
2. **EAN-8**: Compressed version for small items.
3. **UPC-A**: Universal Product Code used in the US and Canada.
4. **UPC-E**: A compressed version of UPC.
5. **Interleaved 2 of 5**: Used in industry, air freight, and other applications.

6. **Codebar**: Used in libraries and sometimes blood banks.
7. **Plessey**: An older code used in retail shelf labeling.
8. **MSI**: A variation of Plessey, used in the US.
9. **PostNet**: Used by the US Postal Service.
10. **Industrial 2 of 5**: An older standard, no longer widely used.
11. **Code 128**: High capacity and reliability, widely used.
12. **Logmars**: The same as Code 39, used in US government specifications.

1.2.2 Two-Dimensional Encoding

There are 5 types:

1. **PDF 417**: High-capacity barcode.
2. **Data Matrix**: Stores a large amount of data in a small space.
3. **Maxi Code**: Used by the US Postal Service for automated sorting.
4. **QR Code**: Explanation follows in the next sections.
5. **Data Code**, **Code 44**, **16K**: Additional types.

1.3 Barcode Standards and Identification Cards

- **EAN** is used for ISBN numbers (books and publications).
- **ISSN SISAC**: International standard serial numbers.
- **OPC**: Used in optical retail.
- **IFT-14**: Container shipping code.
- **Co-Operative Labels**: Managed under specific software systems.

1.4 Barcode Creation Methods

Barcodes can be created using the following methods:

1. **Specialized Software:** These programs generate barcodes from user-defined information and allow exporting them in various formats.

2. **Plugins for Graphic Software:** Examples include *In-Plug* for Illustrator, Photoshop, CorelDRAW, and other similar programs.
3. **Online Barcode Generators:** Users enter the required information and barcode type, and the site generates an image that can be downloaded in different formats.
4. **Encoding Devices:** These devices encode data, information, and numbers into bars, spaces, and lighting variations.

1.5 How to Read a Barcode

Barcodes are read using a **barcode scanner (bar-coded reader)**. It is an optical electronic scanner that projects a laser beam onto the barcode. The beam reflects back from the white lines but is absorbed by the black lines, which do not reflect the light.

A light sensor inside the scanner detects the reflected beam and analyzes it. The processed data is then sent to a computer, which decodes the barcode, retrieving all the associated information such as product name, price, available quantity, and other details.

Figure 1.2: Barcode Reader in Action

1.6 Structure of a Barcode

1.6.1 1D Barcodes (Linear Barcodes)

1D barcodes represent data as a series of vertical bars of varying widths. The data is encoded through the specific arrangement of these bars, and they are widely used in product identification.

A typical structure of a 1D barcode includes:

- **Quiet Zone**: This is an area of blank space around the barcode that ensures the scanner can detect the beginning and end of the code without interference.
- **Start/Stop Characters**: These are special symbols used at the beginning and end of the barcode to indicate where the data starts and stops. For example, in the UPC code, the start/stop character is typically a specific pattern of bars that signals the start and end of the barcode.
- **Data Encoding**: The actual data is encoded as a series of bars and spaces. In a linear barcode, these bars and spaces represent numbers, letters, or other data in a format predefined by the specific barcode standard. Different types of barcodes (e.g., UPC, EAN, Code 39) use different encoding schemes.
- **Check Digit**: This is a calculated digit placed at the end of the barcode. The check digit is used to verify the integrity of the scanned data and to catch errors during the scanning process. It's generated through a mathematical algorithm, such as modulo or weighted sum methods, based on the other digits in the barcode.

1.6.2 2D Barcodes (e.g., QR Codes)

2D barcodes, such as QR codes, store data in both the horizontal and vertical directions, allowing for much more information to be stored in a smaller space.

A typical structure of a 2D barcode includes:

- **Finder Patterns**: These are three large squares located at three corners of the QR code. They help scanners detect the orientation of the barcode.
- **Alignment Patterns**: Smaller squares that help correct distortion in the scanning process and ensure the barcode is read accurately, even if it's tilted or deformed.
- **Data Cells**: The QR code contains a matrix of black and white cells that represent the encoded data. The data is stored in these cells, arranged in a grid format.
- **Error Correction**: Error Correction: QR codes use algorithms such as Reed-Solomon to recover data even if part of the code is damaged or missing.
- **Quiet Zone**: Like 1D barcodes, QR codes also require a quiet zone around the code to help with proper scanning.

1.7 Mathematics of Data Encoding in Barcodes

1.7.1 Numeric Encoding

Barcodes use different encoding schemes to convert data into a sequence of bars and spaces. The most common encoding method for barcodes such as UPC and EAN is **numeric encoding**, where numbers are converted into binary patterns.

For example, in the UPC encoding scheme, each digit (0-9) is represented by a 7-bit binary pattern. Each binary digit is then mapped to a specific bar width in the barcode.

Example 1.7.1. – The number "12345" might be encoded into a binary pattern like:

- ∗ **1** → 1011101
- ∗ **2** → 1110100
- ∗ **3** → 1001101
- ∗ **4** → 1011010
- ∗ **5** → 1101011

These binary patterns are then translated into bars of different widths to form the final barcode.

1.7.2 Alphanumeric Encoding

In some barcodes, such as Code 39, both numbers and letters can be encoded. For example, each alphanumeric character is represented by a sequence of bars and spaces based on a defined encoding scheme. The data is usually encoded into a combination of 9 elements (5 bars and 4 spaces) representing each character.

1.7.3 Error Detection and Correction

For both 1D and 2D barcodes, mathematical algorithms are used to ensure data integrity:

- **Check Digit**: As previously mentioned, the check digit is a calculated digit that ensures the barcode has been scanned correctly. It is computed using various mathematical methods such as:
  - **Modulo 10 (Luhn Algorithm)**: Often used in credit card numbers and UPC codes.
  - **Modulo 11**: Used in some barcodes like ISBN.
  - **Weighted Sum**: The digits in the barcode are weighted and summed, and the remainder when divided by a certain value becomes the check digit.
- **Error Correction** in 2D Barcodes: QR codes, for example, use **Reed-Solomon error correction**, which allows a QR code to still be read even if parts of it are damaged or missing.

These encoding and error correction schemes are essential to ensure that barcodes are reliable and resilient to common scanning issues such as distortion, dirt, or damage.

Chapter

2

One-Dimensional Code

2.1 The Definition and History of Barcodes

Definition 2.1.1. A barcode is an image that uses lines of different thickness to represent information, usually, but not exclusively, digits. It is designed to be read by a special scanner in order to input it to a computer in a quick, automatic way. In this chapter, we will present types of barcodes which are discrete and continuous.

Definition 2.1.2. A barcode is an image that uses lines of different thickness to represent information, usually, but not exclusively, digits. It is designed to be read by a special scanner in order to input it to a computer in a quick, automatic way. In this chapter, we will present types of barcodes which are discrete and continuous.

The first person to invent the barcode was Max Badck in the year 1880 AD, but this project did not see the light. In 1932, a graduate student named Wallace Flint wrote research (automated grocery) at Harvard Business School, in which he explained the use of an automated system.

In 1948, Bernard Silver, a graduate student from Drexel Institute of Technology, in collaboration with his friends Norman Joseph and Norman Johansen, developed the first system that works with ultraviolet ink. For one of the chain stores in Philadelphia to read the products at checkout, but due to the cost of this system, the project failed. Woodland then worked on developing the system and reducing its cost and patented his invention on October 7, 1952. This invention was widely successful, and in 1974, a pack of gum was introduced to a grocery store as the first product in history to use the barcode system.

Codabar is a linear barcode symbology developed in 1972 by Pitney Bowes Corp. It and its variants are also known as Codabar, Ames Code, NW-7, Monarch, Code 2 of 7, Rationalized Codabar, ANSI/AIM BC3-1995 or USD-4. Although Codabar has not been registered for US federal trademark status, its hyphenated variant Code-a-bar is a registered trademark.

2.2 Method of Encoding of 1D Barcode

In this barcode, each digit is encoded using: three narrow lines, a single wide line, two narrow gaps, and a single wide gap. For each digit, these lines and gaps are placed in a different order.

- The digit 0

Example is encoded: narrow line, narrow gap, narrow line, narrow gap, narrow line, wide gap, and wide line.

- **The digit 1**
  Example is encoded: narrow line, narrow gap, narrow line, narrow gap, wide line, wide gap, narrow line.

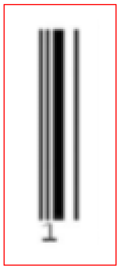


Figure 2.1: Codabar encoding for digits 0 to 9

This method of encoding is called *Codabar*, and the figure below shows how to encode all the numbers from 0 to 9.

1234567895

Figure 2.2: Codabar encoding for digits 0 to 9

Universal Product Code (UPC)

Most countries use 1D barcodes. One of the methods for encoding 1D barcodes is called UPC (Universal Product Code).

This method encodes a twelve-digit number and is referred to as UPC-12 to distinguish it from the six-digit UPC-E barcode.

The universal product code (UPC) is a 12-digit number and associated machine-readable barcode used to identify products being purchased in grocery stores. UPCs encode an individual product, but not its price (this part is done by a store's computer after reading the product identifier). The UPC is maintained by the Uniform Code Council of Dayton, Ohio.

2.2.1 Composition of UPC-12:

The encoding is done as follows:

- a starting symbol
- six encoded digits
- a middle symbol
- six encoded digits
- an end symbol

The start, middle, and end symbols are known as: “Guard Bars."

- The first and last digits are separated from the others and written in a smaller font size.
- The beginning and end symbols are two long narrow lines with a narrow gap between them.
- The middle symbol is a narrow gap followed by a narrow line, then a narrow gap, a narrow line, and a final narrow gap.

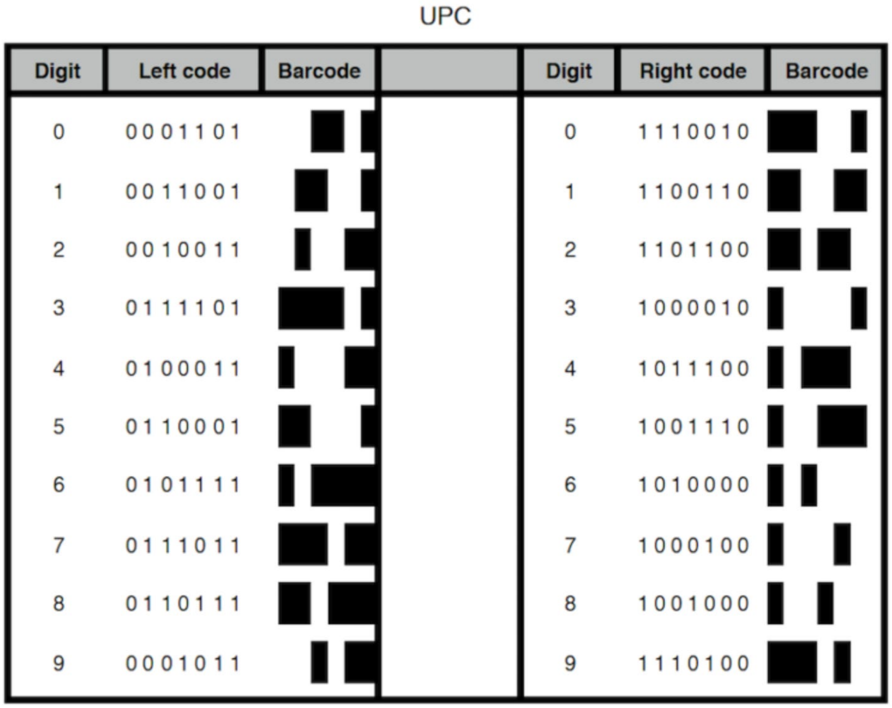
Example 2.2.1. The digits in UPC are encoded into a sequence of lines and gaps in the following manner:

SCAN ME

Figure 2.3: Relizane University Website

2.2.2 UPC Table

Two tables are used, each one having ten rows and seven columns:



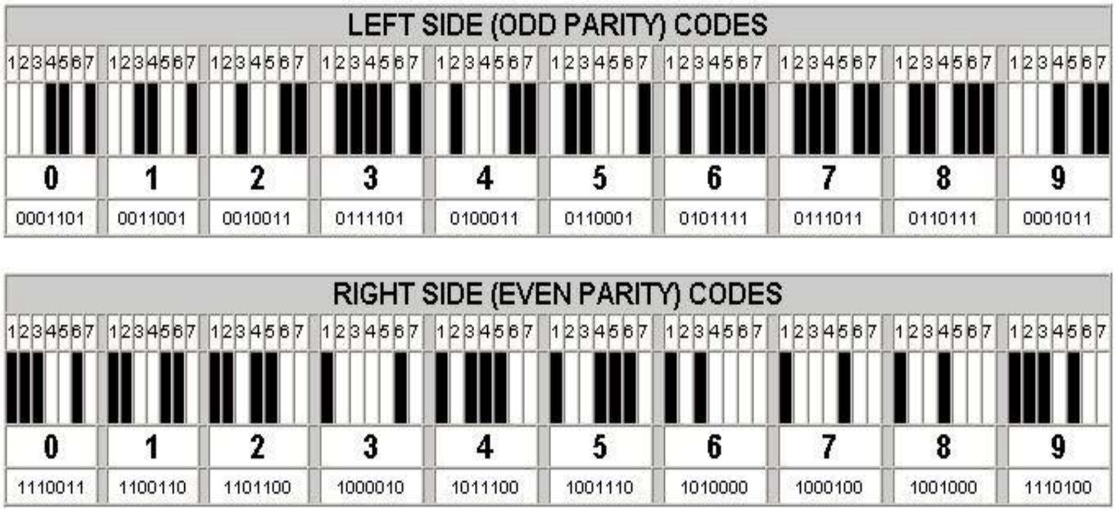
UPC

Digit	Left code	Barcode		Digit	Right code	Barcode
0	0001101			0	1110010	
1	0011001			1	1100110	
2	0010011			2	1101100	
3	0111101			3	1000010	
4	0100011			4	1011100	
5	0110001			5	1001110	
6	0101111			6	1010000	
7	0111011			7	1000100	
8	0110111			8	1001000	
9	0001011			9	1110100	

Figure 2.4: UPC Table

In each table, the rows indicate the coding of digits from 0 to 9. The difference between the two tables is that one table indicates the coding of the left-hand digits, while the other is for the right-hand ones.

In each column the digit 0 represents a gap, and 1 represents a line. This way, the seven digits (0 or 1) in each row in the table make up a sequence of lines and gaps which are that digit's code. If we take, for example, a sequence of three 1s (111), the line would be three times as thick as a line coded from a single 1 digit (010). Another example: 00 stands for a double gap. A single gap is encoded with a single 0.



LEFT SIDE (ODD PARITY) CODES

1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567
0	1	2	3	4	5	6	7	8	9
0001101	0011001	0010011	0111101	0100011	0110001	0101111	0111011	0110111	0001011

RIGHT SIDE (EVEN PARITY) CODES

1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567
0	1	2	3	4	5	6	7	8	9
1110011	1100110	1101100	1000010	1011100	1001110	1010000	1000100	1001000	1110100

Example 2.2.2. Here is the UPC coding for 012345678912:

0 12345 67891 2

Figure 2.5: UPC coding for 012345678912

2.2.3 How to Calculate the Check Digit of a Barcode

The check digit of a barcode is usually calculated automatically by software, but here we demonstrate how to manually compute it to verify the barcode's validity.

For example, in the **EAN-13** system, let's assume the barcode contains the following digits:

400763000011

And suppose the check digit is unknown. We proceed with the following steps:

1. Multiply each digit of the barcode by the following weights in order: **1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3**
2. The results of the multiplications are:

$$\begin{gathered} 4\times 1=4, \quad 0\times 3=0, \quad 0\times 1=0, \quad 7\times 3=21, \quad 6\times 1=6, \quad 3\times 3=9, \\ 0\times 1=0, \quad 0\times 3=0, \quad 0\times 1=0, \quad 0\times 3=0, \quad 1\times 1=1, \quad 1\times 3=3 \end{gathered}$$

3. Add all the results together:

$$4+0+0+21+6+9+0+0+0+0+1+3=44$$

4. Compute the check digit using the formula:

$$\text{Check Digit} = (10-(44 \mod 10)) \mod 10 = (10-4) \mod 10 = 6$$

Therefore, the correct check digit is: **6**

Step-by-step method to verify the check digit of a barcode manually:

1. Suppose we are given the barcode number:

$$639382000393$$

2. We take the first 12 digits only, as the 13th digit is the one we want to calculate to verify the encoding:

$$639382000393 \Rightarrow 63938200039$$

3. Add all the digits in the **odd positions** (1st, 3rd, 5th, etc.):

$$6+9+8+0+0+9=32$$

4. Multiply this sum by 3:

$$32\times 3=96$$

5. Add all the digits in the **even positions** (2nd, 4th, 6th, etc.):

$$3+3+2+0+3=11$$

6. Add the two totals together:

$$96+11=107$$

7. Find the smallest number divisible by 10 that is greater than or equal to the result (i.e., the next multiple of 10):

$$110$$

8. Subtract the result from the next multiple of 10:

$$110 - 107 = 3$$

9. So, the **Check Digit** is:

$$\boxed{3}$$

This is how we confirm that the check digit for the barcode `639382000393` is valid.

Example 2.2.3. We verify that the following UPC barcode is valid: `076301722125`.

Let the barcode be denoted as:

$$a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11}a_{12} = 076301722125$$

- Add all digits in the **odd positions**:

$$0 + 6 + 0 + 7 + 2 + 2 = 17$$

- Multiply the result by 3:

$$17 \times 3 = 51$$

- Add all digits in the **even positions**:

$$7 + 3 + 1 + 2 + 1 + 5 = 19$$

- Add the two results:

$$51 + 19 = 70$$

Since $70 \equiv 0 \pmod{10}$, the check digit is valid.

Example 2.2.4. In the barcode `09948240697`a_{12}, the unknown check digit a_{12} must satisfy the following:

$$3(0 + 9 + 8 + 4 + 6 + 7) + (9 + 4 + 2 + 0 + 9 + a_{12}) = 126 + a_{12}$$

This sum must be divisible by 10. Therefore:

$$a_{12} = 4$$

Example 2.2.5. Consider the barcode `048500001028`, where the check digit is computed as:

$$a_{12} = 10 - \{[3(0 + 8 + 0 + 0 + 1 + 2) + (4 + 5 + 0 + 0 + 0)] \bmod 10\} \bmod 10$$

$$= 10 - [42 \bmod 10] \bmod 10 = 10 - 2 = \boxed{8}$$

Thus, the expected check digit is 8.

Method for Calculating OPC and Interleaved 2 of 5

1. Start from the leftmost digit and apply multiplication according to the Interleaved 2 of 5 standard (Model 212):
   - Multiply the 1st digit by 2
   - Multiply the 2nd digit by 1
   - Multiply the 3rd digit by 2
   - Multiply the 4th digit by 1
   - Continue alternating...
2. Add up the results of all multiplications. If the number consists of two digits, just sum the number as it is.
3. Divide the total sum by 10.
4. Subtract the remainder from 10. If the result is 10, the check digit is 0.

Example 2.2.6. Given code: **020711721**

$$\begin{aligned} 0 \times 2 &= 0 \\ 2 \times 1 &= 2 \\ 0 \times 2 &= 0 \\ 7 \times 1 &= 7 \\ 1 \times 2 &= 2 \\ 1 \times 1 &= 1 \\ 7 \times 2 &= 14 \\ 2 \times 1 &= 2 \\ 1 \times 2 &= 2 \end{aligned}$$

Sum: $0 + 2 + 0 + 7 + 2 + 1 + 14 + 2 + 2 = 30$

$$30 : 10 = 3 \text{ remainder } 0$$
$$10 - 0 = 10 \Rightarrow \text{Check Digit} = 0$$

Check digit $= \mathbf{0}$, thus the code is valid.

2.3 European Article Number (EAN-13)

2.3.1 Composition of (EAN-13)

The 13-digit EAN-13 number consists of four components:

- GS1 prefix - 3 digits
- Manufacturer code - variable
- Product code - variable length
- Check digit

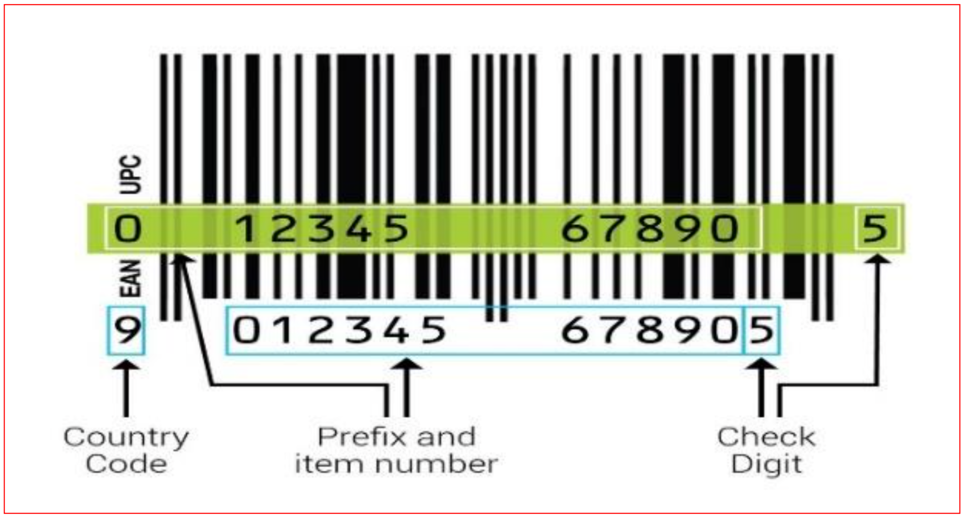


Figure 2.6: Composition of (EAN-13)

2.3.2 Check Digit Calculation for EAN-13 Barcode:

The EAN-13 barcode typically consists of 13 digits:

$$a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11}a_{12}a_{13}$$

- The first 12 digits contain product information.
- The 13th digit, a_{13}, is the check digit used for error detection.

Validation Rule:

A valid EAN-13 barcode must satisfy the following condition:

$$a_1 + 3a_2 + a_3 + 3a_4 + \cdots + 3a_{12} + a_{13} \equiv 0 \pmod{10}$$

Check Digit Formula:

The check digit a_{13} is calculated using the first 12 digits as follows:

$$a_{13} = \left(10 - \left[\left(\sum_{\substack{i=1\\ i \text{ odd}}}^{11} a_i + 3\sum_{\substack{i=2\\ i \text{ even}}}^{12} a_i\right) \bmod 10\right]\right) \bmod 10$$

Explanation:

- Sum all digits at **odd positions** from a_1 to a_{11}.
- Multiply the sum of **even-position digits** (from a_2 to a_{12}) by 3.
- Add both results and take modulo 10.
- Subtract the result from 10 and take modulo 10 again to get the check digit a_{13}.

Example 2.3.1. We verify that the following barcode EAN-13 is valid since:

$$9426281018140368$$

- Add all digits in even positions:
$$2 + 1 + 1 + 1 + 6 = 11$$
- Multiply the result by 3:
$$11 \times 3 = 33$$
- Sum all digits in odd positions:
$$6 + 8 + 0 + 8 + 4 + 3 + 8 = 37$$
- Add the results of step 2 and 3:
$$33 + 37 = 70$$
- Since $70 \equiv 0 \mod 10$, the code is valid.

Example 2.3.2. We verify that the following barcode EAN-13 is valid. Let the barcode be denoted as:

$$a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11}a_{12}a_{13} = 9421023610112$$

We have:

$$(9 + 2 + 0 + 3 + 1 + 1 + 2) + 3(4 + 1 + 2 + 6 + 0 + 1) = 13 + 3 \times 14 = 60 \equiv 0 \mod 10.$$

Example 2.3.3. In the following EAN-13 barcode, we have the check digit $a_{13} = 2$. We can verify that:

4070071967072

$$a_{13} = 10 - [(4 + 7 + 0 + 1 + 6 + 0) + 3(0 + 0 + 7 + 9 + 7 + 7)] \mod 10$$

$$= 10 - [18 + 3 \times 30] \mod 10 = 10 - 108 \mod 10 = 10 - 8 = 2.$$

Thus, the check digit is correct.

2.3.3 Weights for EAN-13 code:

The weight at a specific position in the EAN code is alternating (3 or 1) in a way that the final data digit has a weight of 3 (and thus the check digit has a weight of 1). Codes meet the following rule: Numbering the positions from the right (code aligned to the right), the odd data digits always have a weight of 3 and the even data digits always have a weight of 1, regardless of the length of the code.

Position	1	2	3	4	5	6	7	8	9	10	11	12
Weight	1	3	1	3	1	3	1	3	1	3	1	3

Example 2.3.4. EAN-13 barcode 400638133393a_{13}, where a_{13} is the unknown check digit, the check digit calculation is:

Position	1	2	3	4	5	6	7	8	9	10	11	12
First 12 digit barcode	4	0	0	6	3	8	1	3	3	3	9	3
Weight	1	3	1	3	1	3	1	3	1	3	1	3
Partial sum	4	0	0	18	3	24	1	9	3	9	9	9

The nearest multiple of 10 that is equal to or higher than the checksum is 90.

Subtract them: $90 - 89 = 1$, which is the check digit a_{13} of the barcode.

2.3.4 Binary encoding of data digits into EAN-13 barcode:

EAN-13 uses encoding. The encoded data is usually repeated in plain text below the barcode.

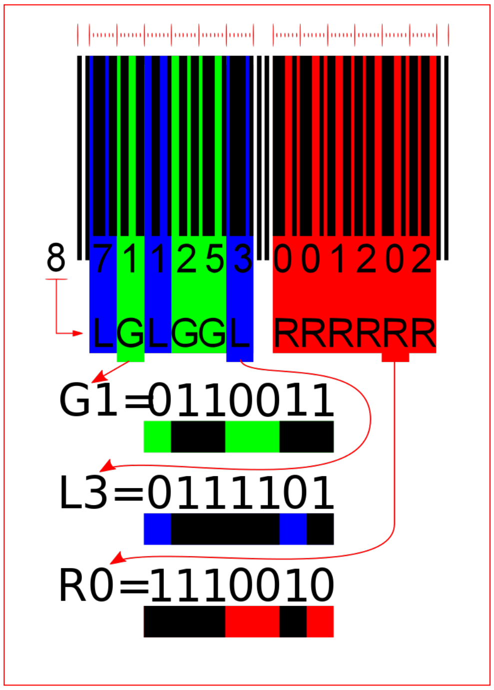


Figure 2.7: Example of EAN-13 Barcode Encoding

The barcode consists of 95 areas (also called modules) of equal width.

Each area can be either white (represented here as 0) or black (represented as 1).

From left to right:

- 3 areas for the start marker (101)
- 42 areas (seven per digit) to encode digits 2-7, and to encode digit 1 indirectly.
- 5 areas for the center marker (01010)
- 42 areas (seven per digit) to encode digits 8-13
- 3 areas for the end marker (101)

Encoding of the digits to encode the 13-digit EAN-13 number, the digits are split into 3 groups: the first digit, the first group of 6, and the last group of 6.

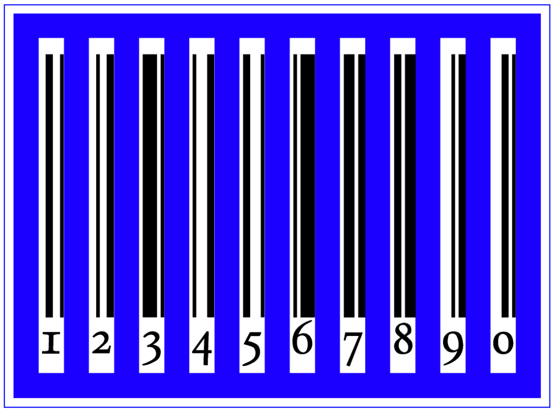


Figure 2.8: Encoding L-digits

The first group of 6 is encoded using a pattern whereby each digit has two possible encodings, one with even parity (denoted with letter G) and one with odd parity (denoted with letter L).

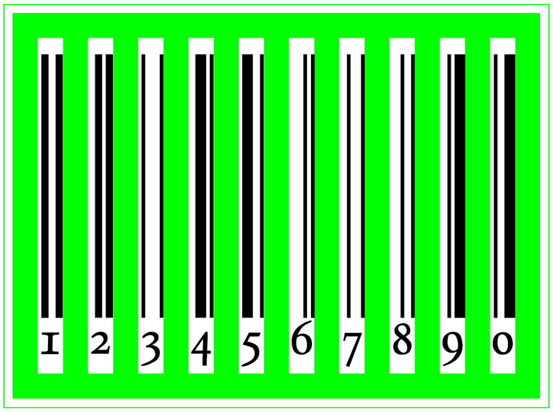


Figure 2.9: Encoding G-digits

The first digit is not represented directly by a pattern of bars and spaces, but is encoded indirectly, by selecting a pattern of choices between these two encodings for the first group of 6 digits, according to the table below.

Figure 2.10: Encoding R-digits

All digits in the last group of 6 digits are encoded using a single pattern RRRRRR, the one also used for UPC.

If the first digit is zero, all digits in the first group of 6 are encoded using the pattern LLLLLL, which is used for UPC. Therefore, a UPC barcode is also an EAN-13 barcode with the first digit set to zero.

First digit	First group of 6 digits	Last group of 6 digits
0	LLLLLL	RRRRRR
1	LLGLGG	RRRRRR
2	LLGGLG	RRRRRR
3	LLGGGL	RRRRRR
4	LGLLGG	RRRRRR
5	LGGLLG	RRRRRR
6	LGGGLL	RRRRRR
7	LGLGLG	RRRRRR
8	LGLGGL	RRRRRR
9	LGGLGL	RRRRRR

2.4 Enlarge and reduce the size of the barcode:

A barcode is a technology used to visually represent data, either numerically or as text. Barcodes are widely used in commercial and industrial systems for efficient data encoding. When printing or using barcodes in scanning systems, adjusting the barcode size may be required to meet specific printing or scanning needs.

2.4.1 How Does a Barcode Work?

A barcode consists of black and white lines representing numbers or characters, which can be read using scanners. The barcode is made up of **modules**—small units that are either black (1) or white (0). When scaling the barcode, the **aspect ratio** between width and height must be maintained to ensure proper scanning.

2.4.2 Scaling Up the Barcode:

Scaling up a barcode requires maintaining the **constant ratio between width and height** of the modules. Scaling should typically occur within the range of 80% to 200% of the nominal size.

2.4.3 Reasons for Scaling Up:

- To use it on larger materials (e.g., large labels or packaging)
- To improve readability on low-quality prints

- To increase scanning distance between the barcode and scanner

123456789012

Figure 2.11: Scaled EAN-13 Barcode (150%)

Example 2.4.1.

2.4.4 Scaling Down the Barcode

Scaling down the barcode may be necessary to fit into a smaller space, such as for small products or tags.

Reasons for Scaling Down:

- To reduce space on packaging or products
- To fit smaller items such as mechanical parts
- To reduce printing costs

Example 2.4.2. If the original EAN-13 barcode size is 5 × 3 cm, scaling it down by 50% results in a new size of 2.5 × 1.5 cm.

6 281234 567895

Figure 2.12: Minimum size (10 %)

6 281234 567895

Figure 2.13: Nominal size (100 %)

6 281234 567895

Figure 2.14: Maximum size (200%)

2.4.5 Risks and Challenges

- Scanning issues when scaled too large or too small
- Loss of detail when excessively scaled down
- Quality loss during reprinting after scaling

2.4.6 Best Practices

- Test the barcode with a scanner after scaling
- Maintain consistent ratios between width and height
- Avoid excessive scaling outside of the permissible range

Chapter

3

Two-Dimensional code

3.1 Definition of QR

There are several types of 2D codes in use by the industry, one of which is QR Code. This chapter provides an overview of QR Code, the standardization activities on this technology and its applications in the various sectors.

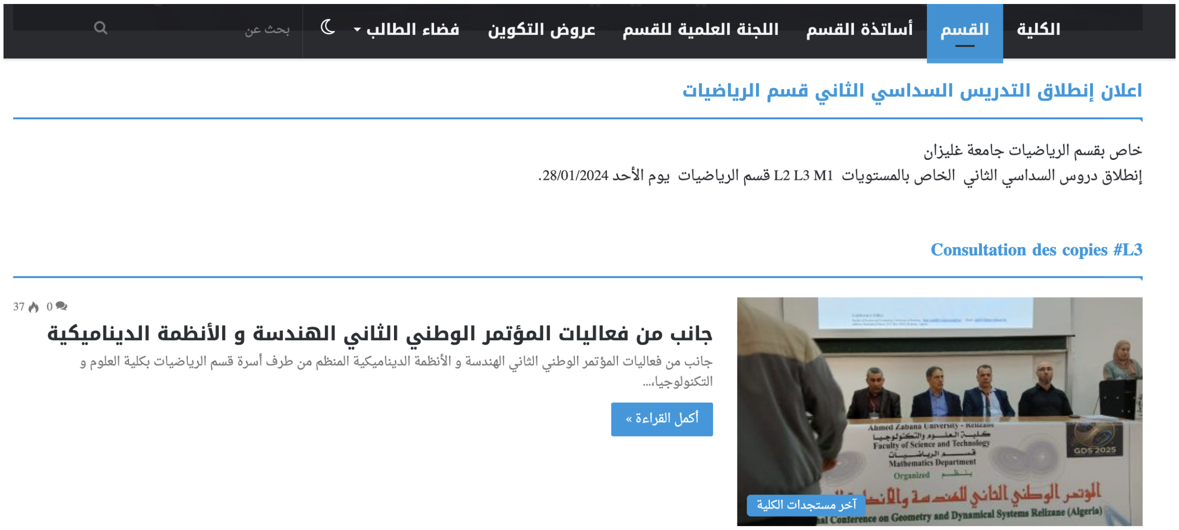
A QR code (an initialism for quick response code) is a type of matrix barcode (or two-dimensional barcode) invented in 1994 by the Japanese automotive company Denso Wave. A barcode is a machinereadable optical label that can contain information about the item to which it is attached. In practice, QR codes often contain data for a locator, identifier, or tracker that points to a website or application. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte/binary, and kanji) to store data efficiently; extensions may also be used. A QR code consists of black (or any other color) squares arranged in a square grid on a white background, which can be read by an imaging device such as a camera, and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data is then extracted from patterns that are present in both horizontal and vertical components of the image.

Example 3.1.1. The following QR is the website of our mathematics Department:

SCAN ME

Figure 3.1: Website of our Mathematics Department

When we scan it with a mobile phone to read it, this page is referred to as WAP.



اعلان إنطلاق التدريس السداسي الثاني قسم الرياضيات

خاص بقسم الرياضيات جامعة غليزان
إنطلاق دروس السداسي الثاني الخاص بالمستويات L2 L3 M1 قسم الرياضيات يوم الأحد 28/01/2024.

Consultation des copies #L3

جانب من فعاليات المؤتمر الوطني الثاني الهندسة و الأنظمة الديناميكية

جانب من فعاليات المؤتمر الوطني الثاني الهندسة و الأنظمة الديناميكية المنظم من طرف أسرة قسم الرياضيات بكلية العلوم و التكنولوجيا،...

Example 3.1.2. The following QR code shows the website of relizane University. When scanned with a mobile phone, it will take you to the website of relizane University.

SCAN ME

Figure 3.2: Website of Relizane University

3.2 The QR structure

The QR consists of five basic elements shown in the image below according to the special color of each part.

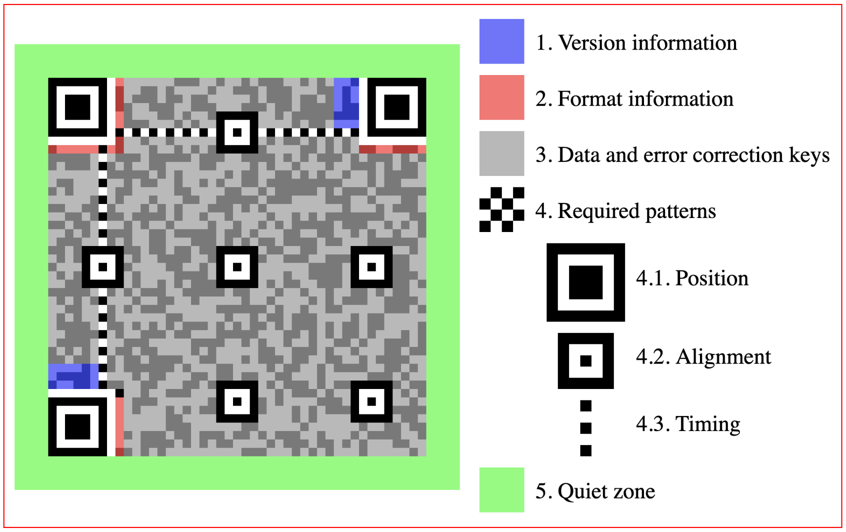


Figure 3.3: QR structure

Unlike the older, one-dimensional barcodes that were designed to be mechanically scanned by a narrow beam of light, a QR code is detected by a 2-dimensional digital image sensor and then digitally analyzed by a programmed processor. The processor locates the three distinctive squares at the corners of the QR code image, using a smaller square (or multiple squares) near the fourth corner to normalize the image for size, orientation, and angle of viewing. The small dots throughout the QR code are then converted to binary numbers and validated with an errorcorrecting algorithm.

3.3 QR versions and dimensions

There are many versions of the QR code starting from 1 to 40, and each version has a dimension defined as follows.

Definition 3.3.1. : To calculate the dimensions of the QR, we multiply 4 by the version number and add 17. It is mean: If we denote the version by and the dimension by d, we have

$$d = 4v + 17$$

. We will include in the following examples different versions.

Example 3.3.2.

- Version 1:

  $$V = 1(21 * 21)$$

  .

  the dimension of QR code version 1 is:

  $$d = 4V + 17$$

  .

  $$(4 * 1) + 17 = 21$$

- Version 2:

  $$V = 2(25 * 25)$$

  .

  the dimension of QR code version 2 is:

  $$d = 4V + 17$$

.

$$(4 * 2) + 17 = 25$$

.

- Version 3:

$$V = 3(29 * 29)$$

.

the dimension of QR code version 3 is:

$$d = 4V + 17$$

.

$$(4 * 3) + 17 = 29$$

.

- Version 10:

$$V = 10(57 * 57)$$

.

the dimension of QR code version 3 is:

$$d = 4V + 17$$

.

$$(4 * 10) + 17 = 57$$

.

- Version 25:

$$V = 25(117 * 117)$$

.

the dimension of QR code version 3 is:

$$d = 4V + 17.$$

.

$$(4 * 25) + 17 = 117$$

.

3.4 The QR storage

This following table shows the Maximum character storage capacity (40-L) Where the character refers to individual values of the input mode/datatype.

Input mode	Max characters	Bits / characters	Possible characters, default encoding
Numeric only	7089	3.33333	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Alphanumeric	4296	5.5	0-9, A-Z (upper case only), space,
Kanji / kana	1817	13	Shift JIS

3.4.1 Polynomials in $\mathbb{F}_{256}$

The amount of data that can be stored in the QR code symbol depends on the datatype (mode, or input character set), version (1, ..., 40, and error correction level). The maximum storage capacities occur for version 40 and error correction level L (low), denoted by 40-L

3.5 The QR Code Generator

3.5.1 Sequencing data

Let's suppose we want to create a QR code for the string https://www.qrcode.com/ (the official QR code site). First of all, we need to find the correct encoding mode. Each mode has a corresponding value, according to the following table:

	Encoding mode	Value bits
	Numeric	0001
	Alphanumeric	0010
	kanji	1000

Let's aim for the smallest version possible: since it's 23 characters long, we can check in various tables around version 2:

	Error correction level	Numeric mode	Alphanumeric mode	Kanji mode
	Low (L)	77	47	20
	Medium (M)	63	38	16
	Quartile (Q)	48	29	12
	High (H)	34	20	8

The first 4 bits of our data sequence are 0100, our encoding mode. we'll need 8 bits, and 23 (the length of our string) in binary is 10111, our first bits are 01000001 0111 .

Now convert everything to binary and concatenate it to the previous sequence:

	01000001	01110110	10000111	01000111	01000111
	00000111	00110011	10100010	11110010	11110111
	01110111	01110111	01110010	11100111	00010111
	00100110	00110110	11110110	0100110	01010010
	11100110	00110110	11110110	11010010	1111.....

Now we have to put a termination block, which is exactly 4 zeroes, so the last codewords will be 11110000. We have filled all the 8 bits of the last codeword, otherwise we'd have to fill the remaining bits with zeroes.

3.5.2 Error correction

Most of the Math in QR codes in performed in the Galois Field of order$2^8 = 256$.In this set, denoted as $\mathbb{F}_{256}$.

The algorithm chosen for EDC in QR codes is the Reed-Solomon error correction, which is widely used for streaming data (e.g. CDs, wireless communications) because it allows to correct errors found in bursts, rather than single isolated cases. Operations on $\mathbb{F}_{256}$: The "addition'. The neutral element with relation to is still 0, as a + 0 = a. Also, every element is the opposite of itself, since $a = -a$ and since "subtraction" is defined as adding the opposite of the second term, this also means that the subtraction" is equivalent of the "addition"! In fact:

$$a - b = a + (-b) = a + b$$

Now, about the multiplication. A Galois Field is cyclic, meaning that every non zero element can be expressed as the power of a "primitive element" a. So, inF_{256} , if a = a^n and b = a^m then $a.b = a^n$ $.a^m = a^n + m$. But, as we said, a Galois Field is cyclic, so $a^{256} = a$. This means that we can take the exponent $(n+m)$ mod $255 so, we can simplify our computations a bit. In the end, a \cdot b = a^{(n+m) \bmod 255}$ (if both a and b are non-zero; the result is of course 0 otherwise) This also means that for every a ,$a^{256} = a$, and then $a^{255} = 1$, therefore $a^{254} = a^{-1}$ i.e. is the inverse of a. So now we have a way to do divisions:

$$\frac{a}{b} = \frac{a^n}{a^m} = a^n (a^m)^{254} = a^{(n+m\cdot 254)} \quad \bmod 255$$

But how will that serve us for error correction? Let's see. ..

3.5.3 Polynomials in F_{256}

The Reed-Solomon algorithm uses polynomials, and have this form:

$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$

where $a^0, ..., a^n$ are the coefficients inF_{256}, while x is the variable.

Polynomial multiplication :

We can multiply polynomials between them like we used to do with polynomials on real numbers. Suppose we have

$$p_1(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

$$p_2(x) = b_m x^m + b_{m-1} x^{m-1} + \cdots + b_1 x + b_0$$

Take the first term of$p_1(x)$, i.e. $a_n x$, then multiply it with all the terms of $p_2(x)$:

$$a_n x^n \cdot p_2(x) = a_n b_m x^{n+m} + a_n b_{m-1} x^{n+m-1} + \cdots + a_n b_1 x^{n+1} + a_n b_0 x^n$$

Then do the same with the second term of $p_1(x)$, then the third, and so on Finally, sum them all together.

Example 3.5.1.

$$x^2 + 3x + 2 \quad \text{and} \quad 2x^2 + x + 7$$

$$\begin{aligned}(x^2 + 3x + 2)(2x^2 + x + 7) &= x^2(2x^2 + x + 7) + 3x(2x^2 + x + 7) + 2(2x^2 + x + 7) \\ &= 2x^4 + x^3 + 7x^2 + 6x^3 + 3x^2 + 21x + 4x^2 + 2x + 14 \\ &= 2x^4 + (6+1)x^3 + (7+3+4)x^2 + (21+2)x + 14 \\ &= 2x^4 + 7x^3 + 14x^2 + 23x + 14\end{aligned}$$

We end up with a polynomial with 5 terms, which is the sum of the amount of terms of both polynomials, minus 1.

Polynomial divisions

dividend polynomial $4x^3 + 4x^2 + 7x + 5$, and a divisor polynomial $2x + 1$

Basically, these are the steps:

1- divide the first term of the dividend polynomial $4x^3$ with the first term of the divisor ($2x$, and get $2x^2$)
2- multiply the divisor polynomial by the above quotient (you'll get $4x^3 + 2x^2$)

3- get the rest by subtracting the result from the dividend (you'll get $2x^2 + 7x + 5$) 4- if the degree of the rest is lower than the degree of the divisor, you're done otherwise. For the division above (in the field of real numbers), you'll get a polynomial quotient of $2x^2 + x + 3$, and a rest of 2. In code

The quotient polynomial will always be long the difference in length of the dividend and the divisor, plus one. So, we're defining a function that returns only the rest of the division The size of the quotient is needed just to count the steps to do .

3.5.4 Theory of Reed-Solomon

Theory says that a Reed-Solomon error correction data sequence spanning n codewords allows to recover up to n/2 unreadable codewords, they being among the data sequence or in error correction sequence itself

Level	Letter	Data recovery	
Low	L	~7%	
Medium	M	~15%	
Quartile	Q	~25%	
High	H	~30%	

For example, a version 2 QR code contains 44 codewords in total. We want to recover up to 11 (which is 25%) of them, which means that we must reserve 22
From this, it's also clear we can't do much better than level H of error correction. If, for example, we wanted 18 codewords to be recoverable out of 44, then we had to use 36 codewords just for error correction.
we've sequenced our data (the string https://www.qrcode.com/) into an array of bytes Now we've treated polynomials as arrays of values between 0 and 255 And that's handy, since for error correction we have to view our data as a polynomial with the codewords as coefficients .
Basically, we have our data that becomes this polynomial, called the message polynomial:

$$65x^{27} + 118x^{26} + 135x^{25} + 71x^{24} + \cdots + 17x + 236$$

But we have 44 total codewords in our version 2 QR code, so we have to multiply this by x to the power of the error correction codewords, i.e. 16. In the end we have:

$$65x^{43} + 118x^{42} + 135x^{41} + 71x^{40} + \cdots + 17x^{17} + 236x^{16}$$

Now that we have our big polynomial, we have to divide it by... divisor polynomial, and take the rest of this division: the coefficients of the rest polynomial are going to be our error correction codewords !

3.5.5 The generator polynomial

If we have to fill n codewords with error correction data, we need the generator polynomial to be of degree n, so that the rest is of degree n - 1 and so the coefficients are exactly n. What we're going to compute is a polynomial like this:

$$(x - \alpha^0)(x - \alpha^1)(x - \alpha^2) \cdots (x - \alpha^{n-2})(x - \alpha^{n-1})$$

Now, as we've said, in F_{256} subtraction is the same as addition, and we've also chosen α to be 2. Finally, there are 16 codewords for medium correction in a version 2 QR code, so our generator polynomial is this one:

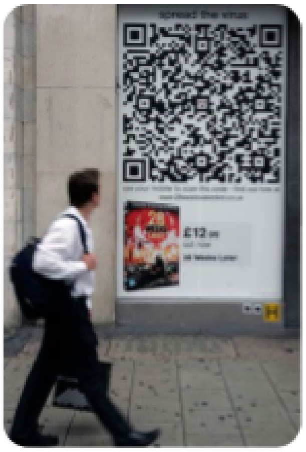
$$\begin{aligned}&(x+1)(x+2)(x+4)(x+8)(x+16)(x+32)(x+64)(x+128)\\&\quad \cdot (x+29)(x+58)(x+116)(x+232)(x+205)(x+135)(x+19)(x+38)\\&= 65x^{43} + 118x^{42} + 135x^{41} + 71x^{40} + \cdots + 17x^{17} + 236x^{16}\end{aligned}$$

3.6 Uses of QR Code

1. You can design your personal card and place your information using a QR Code. This way, people won't need to carry a card all the time. All they need to do is scan the code with their phone camera to save your information directly to their device.

2. Some magazines have started placing QR Codes on certain pages. When you scan them with your device, you can be redirected to an article, a video, or any media the magazine wants to share with you.

3. If you're in a hurry and you spot an advertisement that catches your attention, just take a quick photo of the QR Code on it and review it later at your convenience.



4. Some product shops place QR Codes below the product displays. By scanning the code, you can learn more details about the item. In fact, some stores even allow you to purchase the item online via the code.

5. Some advertisements consist entirely of a QR Code. Once you scan it, you'll find a name, phone number, job title, and the related information from the ad.

6. You can find specific locations on a map using apps that rely on QR Codes.

3.7 List of barcodes for most countries in the world

Canada 019
USA - 000
Canada 030 - 039 USA
France 300 - 379
Bulgaria 380
Slovenija 383
Croatia 385
(BIH) Bosnia-Herzegovina 387
Germany 400 - 440
Japan 450 - 459
Russian Federation 460 - 469
Kyrgyzstan 470
Taiwan 471
Estonia 474
Latvia 475
Aze rbaijan 476
Lithuania 477
Uzbekistan 478
SriLanka 479
Philippines 480
Belarus 481

Ukraine 482
Moldova 484
Armenia 485
Georgia 486
Kazakhstan 487
HongKong 489
Japan 490 - 499
UK 500 - 509
Greece 520
Lebanon 528
Cyprus 529
Macedonia 531
Malta 535
Ireland 539
Belgium , Luxembourg 540 - 549
Portugal 560
Iceland 569
Denmark 570 - 579
Poland 590
Romania 594
Hungary 599
South Africa 600 -601
Bahrain 608
Mauritius 609
Morocco 611
Algeria 613
Tunisia 619
S yria 621
Egypt 622
Libya 624
Jordan 625
Iran 626
Kuwait 627
Saudi Arabia 628
Emirates 629
Finland 640 - 649
China 690 - 695
Norway 700 - 709
Sweden 730 - 739
Guatemala 740
ElSalvador 741
Honduras 742
Costa Rica 744
Panama 745
Republica Dominicana 746

Mexico 750
Canada 754 - 755
Venezuela 759
Schweiz, Suisse, Svizzera 760 - 769
Colombia 770
Uruguay 773
Peru 775
Bolivia 777
Argentina779
Chile 780
Paraguay 784
Ecuador786
Brasil 789 - 790
Italy 800 - 839
Spain 840 - 849
Cuba 850
Slovakia 858
Czech 859
YUSerbia , Montenegro 860
Mongolia 865
North Korea 867
Nether lands870 - 879
South Korea 880
Cambodia 884
Thailand 885
Singapore 888
India 890
Vietnam 893
Indonesia 899
Austria 900 - 919
Australia 930 - 939
NewZealand 940 - 949
Head Office 950
Malaysia 955
Macau 958

Conclusion

Barcodes have become an essential technology in modern data management, logistics, retail, and identification systems. From the simplicity of 1D linear barcodes like UPC and EAN to the complexity and high capacity of 2D codes such as QR codes, barcoding systems offer reliable, fast, and cost-effective solutions for encoding and retrieving data. Understanding the types, structures, and encoding methods behind barcodes reveals the sophisticated mathematics and engineering that support their functionality. Key concepts such as numeric and alphanumeric encoding, check digit verification, error detection, and correction mechanisms ensure both accuracy and integrity of the data. As industries continue to evolve with digital transformation, barcodes remain a bridge between physical objects and digital systems, enabling automation, real-time tracking, and seamless data exchange. The study of barcodes not only provides insight into their practical applications but also highlights the interdisciplinary collaboration between mathematics, computer science, and engineering.

References

1. Ali, M. and Kumar, P. QR Code: Structure and Mathematical Analysis. International Journal of Computer Applications. 22–26 (2018)
2. Bar Code 1: A Web Of Information About Bar Code UPC and EAN Bar Code Page". Adams Communications. June 20, 2013
3. Rosen, Kenneth H. (1993), Elementary Number Theory and its Applications (3rd ed.), Addison-Wesle.
4. Roger C. Palmer, The Bar Code Book: A Comprehensive Guide to Reading, Printing, Specifying, Evaluating, and Using Bar Code and Other Machine-readable Symbols 2007
5. Dauben, Joseph W. (2007), "Chapter 3: Chinese Mathematics", in Katz, Victor J. (ed.), The Mathematics of Egypt, Mesopotamia, China, India and Islam : A Sourcebook, Princeton University Press, pp. 187–384.
6. Tariq Al-Rawi, Encyclopedia of Barcode and QR Code, 2015, First edition.
7. Pisano, Leonardo (2002), Fibonacci's Liber Abaci, translated by Sigler ,Laurence E., Springer-Verlag, pp. 402–403.
8. Dence, Joseph B.; Dence, Thomas P. (1999), Elements of the Theory of Numbers, Academic Press.
9. Generate QR code → `https://hovercode.com/`
10. Generate bar-code → `https://barcode.tec-it.com/fr/`