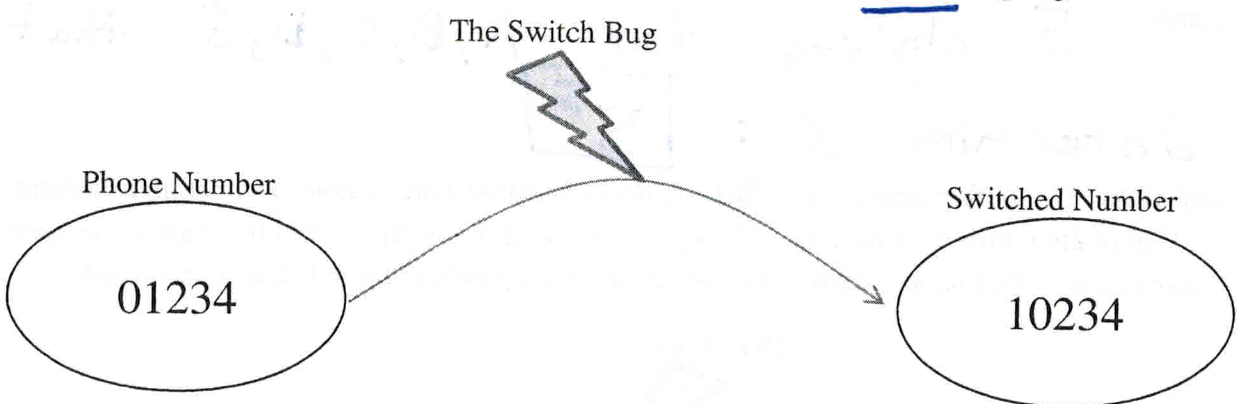
Coding Theory

Warm up problem: Wrong Number. (5.7 of Ecco)

In a city called Five people are getting very upset. In Five, phone numbers are 5 digits long made with the numbers 0 to 4, but lately many phone calls were resulting in wrong numbers. After a correct number has been called, in transmission, one pair of adjacent digits gets switched:



Going against all beliefs about the number five; the city has decided to add a sixth number to their phone system called a *check digit*. This was decided even though the sixth digit will still be prone to switching with the fifth digit. After doing so, called numbers that experience "the switch bug" will result in a nonfunctioning number; the caller will be informed that a switch bug has been detected and to please redial the number.

Help the town of Five by designing a mod 5 congruence equation that picks the check digit for each existing five-digit phone number. The congruence should be true when any new 6-digit phone number is plugged in; but should be false after a switch bug occurs. Which means our solution needs to determine when a switch occurs (*error detection*) but does not need recover the original phone number from the switched number (*error correction*). Phone #'s : ABCDEX

Pick X so $B + D + X \equiv 0 \pmod 5$ TRUE

Suppose we have a switch bug

Say $A \leftrightarrow B$:

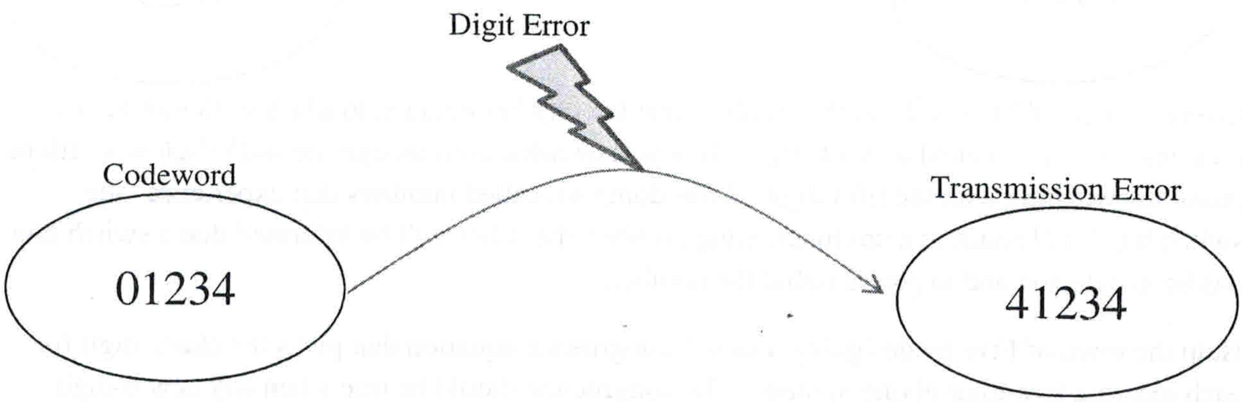
$A \neq B$

$A + D + X \not\equiv B + D + X \equiv 0 \pmod 5$

Definition 1: A *code* is a collection of codewords. A *codeword* is a "word" made of digits or letters or other symbols.

Example 1: In the wrong number problem, the code is the collection of all 6-digit phone numbers while the codewords are the 6-digit phone numbers. How many codewords are there in the code used to solve the wrong number problem? In other words, what is the maximum number of people that could live in the town of Five each having different 6-digit phone numbers? 5 choices for A, B, C, D, E that determine X : $\boxed{5^5}$

Definition 3: Another type of error that can occur is called a *digit error*. Here, a digit becomes a different digit during transmission. It is possible for multiple digit errors to occur in the same transmission. The picture below illustrates a digit error where one symbol was corrupted:



Definition 4: Given two words x, y of the same length the *Hamming distance* $H(x, y)$ is the number of places in which there are different symbols.

$$\begin{matrix} x = 1010101010 \\ y = 1111111111 \end{matrix} \implies H(x,y) = 5$$

$$\begin{matrix} x = AB\$12 \\ y = AC\#12 \end{matrix} \implies H(x,y) = 2$$

$$\begin{matrix} x = 12345 \\ y = 54321 \end{matrix} \implies H(x,y) = 4$$

Note: A larger hamming distance between each codeword allows the code to detect/correct more errors.

Detecting Digit Errors

The sender and receiver team first agrees upon a code containing codewords. They will only send codewords from this code. That way if something other than a codeword (from the agreed upon code) is received an error is immediately detected.

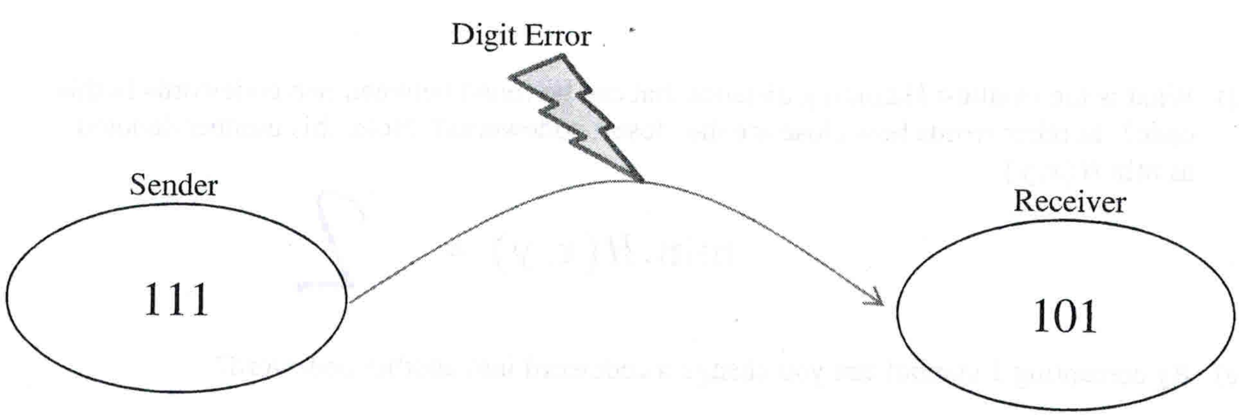
To successfully design a digit error detecting code, all codewords should be chosen so that, transmission errors cannot transform codewords into other codewords.

Example 2: The following binary code is designed to communicated 3 things yes, no, and maybe. Here are the three codewords that make the code:

111	*(yes)*
001	*(no)*
010	*(maybe)*

This example will explain why the "*yes, no, maybe*" code should be used to detect errors over a transmission that is subject a single digit error but not subject to two digit errors.

a) Suppose the sender receiver team experience the following over a transmission that is known to corrupt up to one digit during transmission:



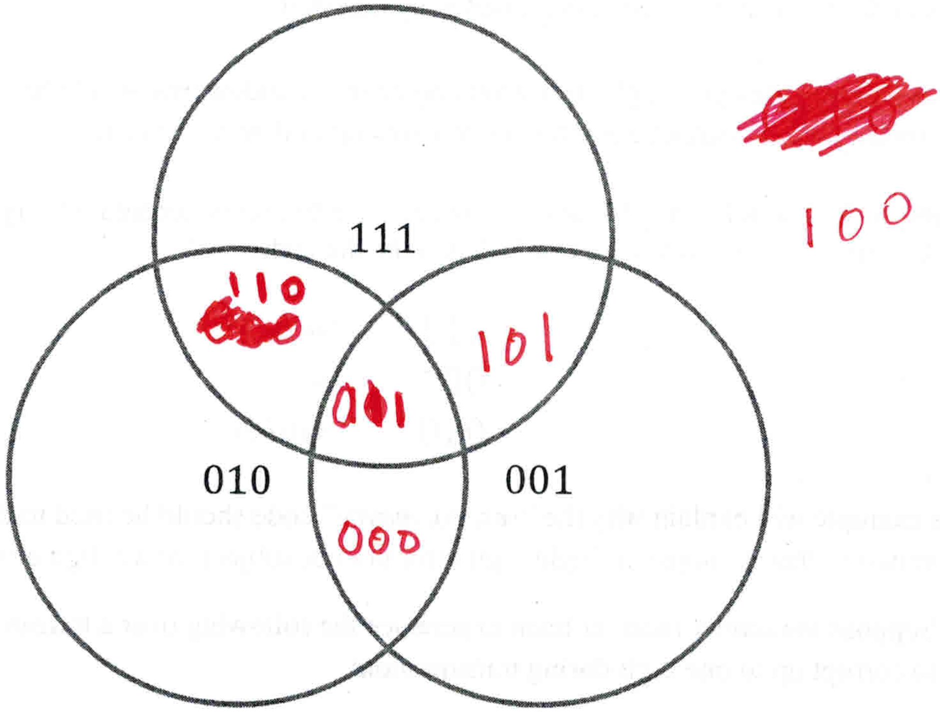
How does the receiver detect the error?

101 is not part of the code

b) In part a) can the receiver correct 101 back to 111?

No, we don't know which codeword was sent by only looking at 101.

c) In the Venn diagram below a circle is drawn around each codeword. The circles represent all transmission errors that are within a hamming distance of 1 from each codeword. Corrupt a single digit in each codeword and place the resulting transmission error in the correct circles.



d) What is the smallest Hamming distance that can be found between two codewords in this code? In other words how close are the closest codewords? Note: this number denoted as min $H(x, y)$.

$$\min H(x, y) = 2$$

e) By corrupting 1 symbol can you change a codeword into another codeword?

No

f) By corrupting 2 symbols can you change a codeword into another codeword?

Yes

g) Up to how many errors can the "*yes, no, maybe*" code detect?

$\boxed{1}$

Correcting Digit Errors

Again the sender and receiver team first agrees upon a code containing codewords. This time, if a transmission error is received the Hamming distance will be found between the transmission error and each codeword. One of the codewords will have the smallest Hamming distance when compared with the transmission error; the transmission error will be corrected to this codeword.

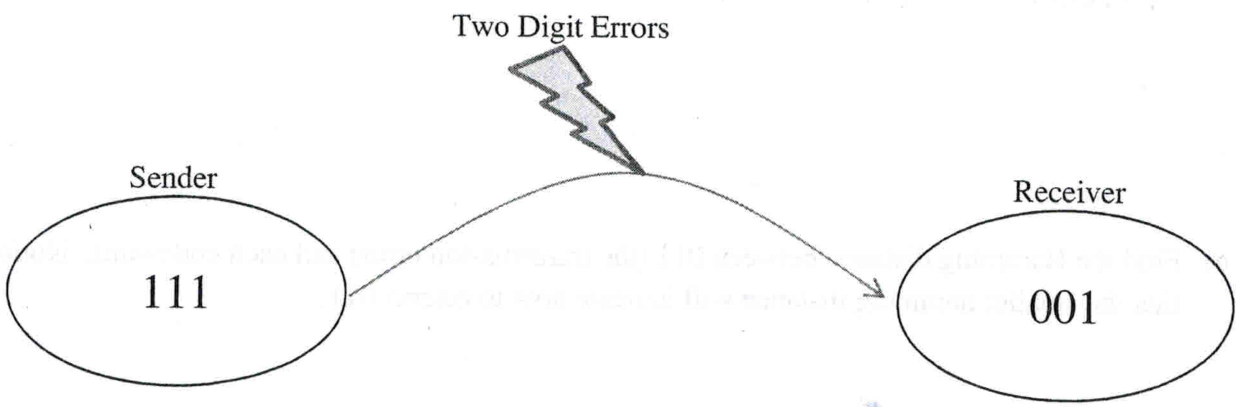
To successfully design an error correcting code, the set of codewords should be chosen so that, transmission errors of any codeword cannot transform into possible transmission errors of any other codeword.

Example 3: The following binary code is designed to communicated 2 things yes and no. Here are the two codewords that make the code:

111 *(yes)*
000 *(no)*

This example will explain why the "*yes, no*" code should be used to **detect** errors over a transmission that is subject to no more than two-digit errors and **correct** errors over a transmission that is subject to no more than one-digit error.

a) Suppose the sender receiver team experience the following over a transmission that is known to corrupt up to two digits during transmission:



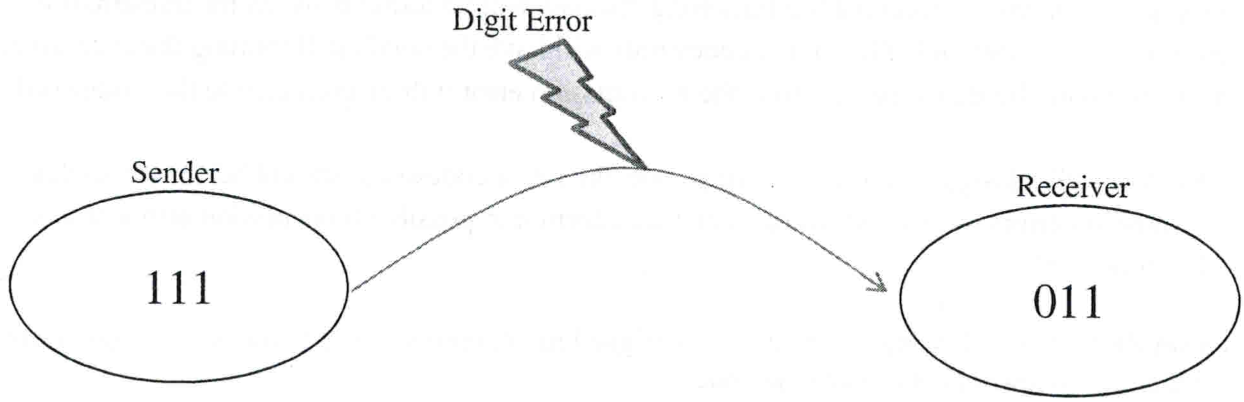
How does the receiver detect the error?

001 is not part of the code

b) In part a) can the receiver correct 001 back to 111?

No, we dont know if 111 or 000 was sent.

c) Suppose the sender receiver team experience the following over a transmission that is known to corrupt up to one digit during transmission:



How does the receiver detect the error?

011 is not part of the code

d) In part c) can the receiver correct 011 back to 111?

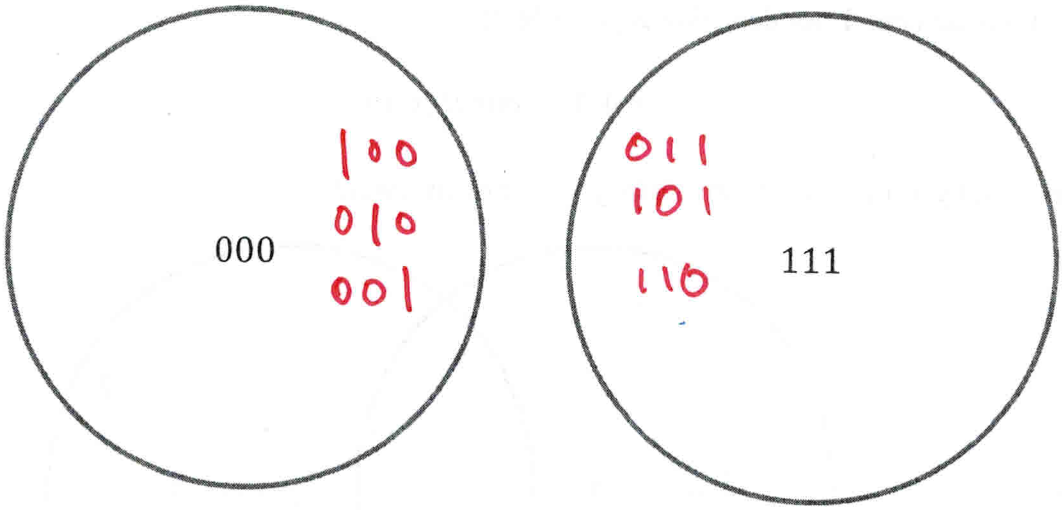
011 is closer to 111 than it is to 000

e) Find the Hamming distance between 011 (the transmission error) and each codeword. Notice that the smaller hamming distance will indicate how to correct 011.

$H(111{,}011) =$ 1

$H(000{,}011) =$ 2

f) In the Venn diagram below a circle is drawn around each codeword. The circles represent all transmission errors that are within a hamming distance of 1 from each codeword. Corrupt a single digit in each codeword and place the resulting transmission error in the correct circles.



g) For this code what is $\min H(x, y)$?

$$\min H(x, y) = 3$$

h) By corrupting 2 symbols can you change a codeword into another codeword?

No

i) By corrupting 3 symbols can you change a codeword into another codeword?

Yes

j) Up to how many errors can the "*yes, no*" code detect?

2

k) If one symbol of a codeword was corrupted during transmission could the receiver determine which codeword was sent? (the receiver can assume that no more than one symbol is corrupt)

Yes

l) If two symbols of a codeword are corrupted during transmission could the receiver determine which codeword was sent? (the receiver can assume that no more than two symbols are corrupt)

No

m) Up to how many errors can the "*yes, no*" code correct?

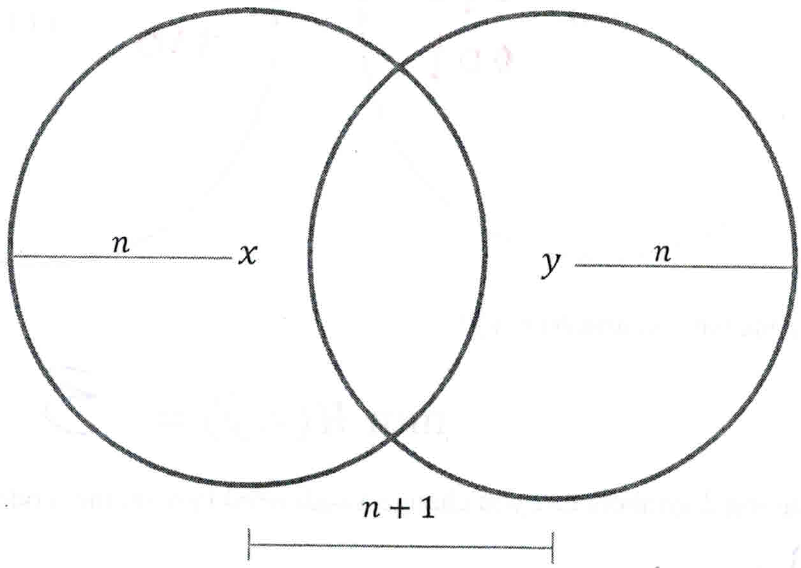
1

Summary of Digit Error Detecting and Correcting

Digit Error Detecting Theorem: Suppose up to n characters of a codeword can be corrupted during transmission. Error detection is possible if:

$$n + 1 \leq \min H(x, y)$$

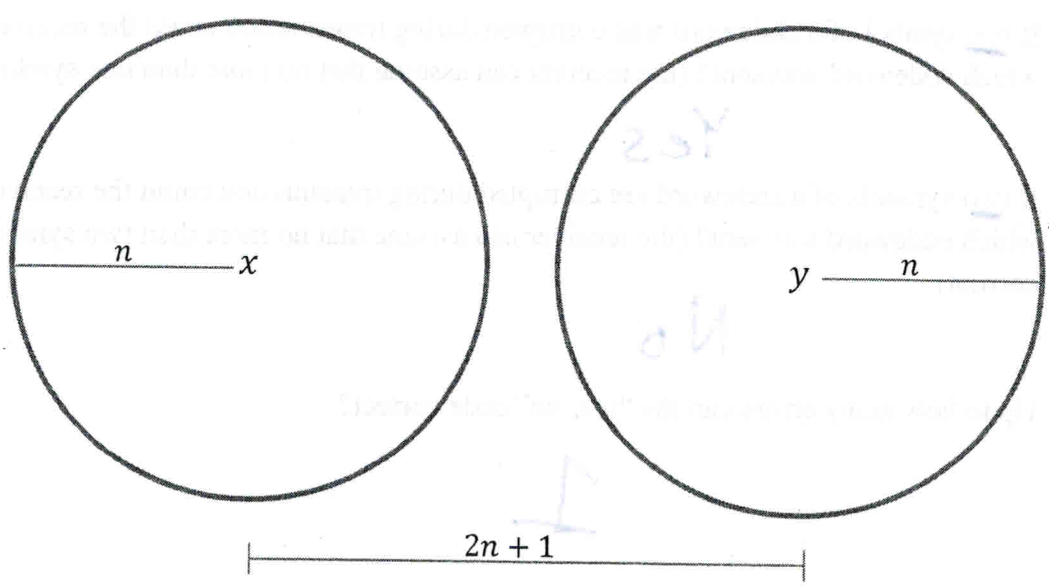
Picture x and y as the closest ($\min H(x, y)$) pair of codewords:



Digit Error Correcting Theorem: Suppose up to n characters of a codeword can be corrupted during transmission. Error correction is possible if:

$$2n + 1 \leq \min H(x, y)$$

Picture x and y as the closest ($\min H(x, y)$) pair of codewords:



ISBN-10 vs ISBN-13

US$12.95

ISBN-10 → ISBN 0-8048-1905-X

ISBN-13 → 9 780804 819053

Note: the only differences between the ISBN-10 and the ISBN-13 are in the first 3 digits and the last digit.

Example 4: The ten-digit International Standard Book Number ISBN-10 was introduced in 1970. The first nine digits $abcdefghi$ are called information digits while the tenth digit j is a check digit chosen so:

$$10a + 9b + 8c + 7d + 6e + 5f + 4g + 3h + 2i + j \equiv 0 \pmod{11}$$

When $j \equiv 10 \pmod{11}$, an X is used instead.

a) What is the $\min H(x, y)$ for the ISBN-10 code?

$cx \equiv cy \Rightarrow x \equiv y \pmod{11}$ $(c \neq 0)$

$\therefore \min H(x,y) > 1$

$H\begin{pmatrix} 00000\ 00000, \\ 10000\ 00001 \end{pmatrix} = 2$ $\therefore \min H(x,y) = \boxed{2}$

b) How many digit errors can the ISBN-10 code detect?

1

c) How many digit errors can the ISBN-10 code correct?

0

d) The thirteen-digit International Standard Book Number ISBN-13 was introduced in 2007. The first twelve digits are information digits while the last digit x_{13} a check digit:

$$\sum_{\substack{i=1 \\ i \text{ is odd}}}^{13} x_i + \sum_{\substack{i=2 \\ i \text{ is even}}}^{12} 3x_i \equiv 0 \pmod{10}$$

What is the min $H(x, y)$ for the ISBN-13 code?

$3x_i \equiv 3x_j \Rightarrow x_i \equiv x_j \pmod{10}$

$\therefore \min H(x,y) > 1$

$H\begin{pmatrix} 00000\ 00000\ 000 \\ 10000\ 00000\ 009 \end{pmatrix} = 2$

$\therefore \min H(x,y) = 2$

e) How many digit errors can the ISBN-13 code detect?

1

f) How many digit errors can the ISBN-13 code correct?

0

g) Can the ISBN-13 code detect the switch bug?

$\boxed{\text{NO}}$

05000 00000 00~~0~~ 5

50000 00000 005

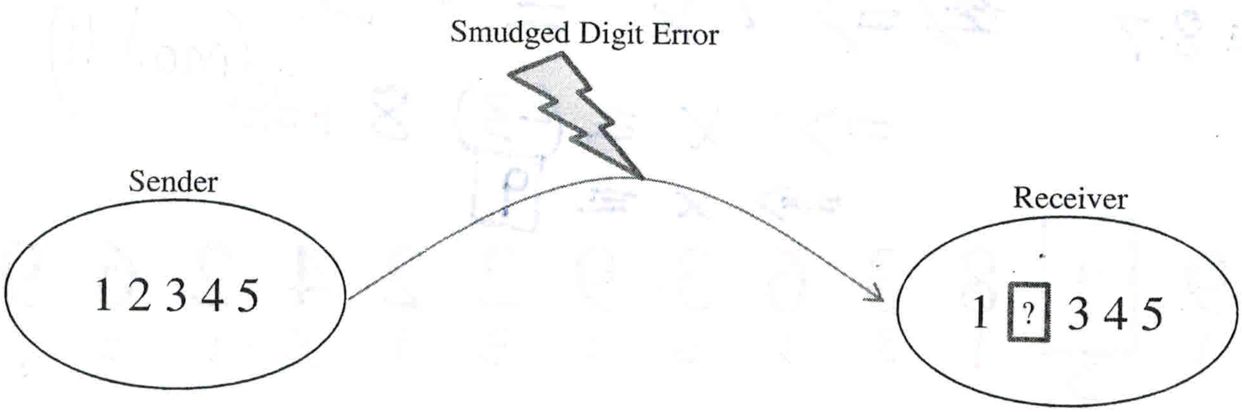
h) Can the ISBN-10 code detect the switch bug? Say $a \leftrightarrow b$, $a \not\equiv b$

$$(10a + 9b) - (10b + 9a) \equiv a - b \not\equiv 0 \pmod{11}$$

$$(10b + 9a) + 8c + \cdots + j \not\equiv (10a + 9b) + 8c + \cdots + j \equiv 0 \pmod{11}$$

$\boxed{\text{Yes}}$

i) Here is another type of error that can occur:



In other words, this is single digit error where the location of the error is known. Can the ISBN-10 correct a smudged digit error?

Yes, we can solve for any variable

j) Can the ISBN-13 correct a smudged digit error?

Yes, as above.

k) Correct the following smudged digit errors:

10	9	8	7	6	5	4	3	2	1
3	6	3	?	2	2	4	2	6	4

$7 \equiv -4$

$\cancel{-3+(-1)}+2+7x+\cancel{1}-\cancel{1}+\cancel{5+6}+1+\cancel{4}$

$\equiv 7x+3 \equiv 0$

$1 \equiv 12$ $+11$

$\equiv (-3)(-4)$

$\equiv 8 \cdot 7$

$\Rightarrow 7x \equiv -3 \pmod{11}$

$\Rightarrow x \equiv (-3)\,8 + 33$

$\Rightarrow x \equiv \boxed{9}$

9	?	8	3	6	3	9	2	2	4	2	6	9
1	3	1	3	1	3	1	3	1	3	1	3	1

$9+3x+8+9+6+9+9+\cancel{6+2+2}+2\cancel{+8}+9$

$5+4+3x \equiv 0$

$\Rightarrow 3x \equiv 1$

$\Rightarrow x \equiv \boxed{7}$

You may be awarded bonus marks by writing the title/author of the book assigned to the above ISBN on your final.