Term Project: Check Digits

A. **Aim.** The aim of this project is to construct a few check digit system checker using MATLAB and use it find the check digits for various types of identification numbers or check if there are any errors.

B. **Instructions.**

1. You must submit your report for this project

   by 6 November, Friday 3.30pm.

2. Your submission will be checked for plagiarism. All parties found involved in **plagiarism will be severely dealt with**. These include
   (a) students who copy from another student's work or other sources;
   (b) students who allow their works to be copied.

3. Your submission must be in **one PDF file**, with **all text selectable**. In other words, when I use the 'Select Tool' in Adobe Reader to select all the text in the PDF file, I should be able to copy all the text in the file, except text in plots produced by MATLAB® and in mathematical formulas produced by short LaTeX codes.
   The reason for this is to allow for successful operation of the plagiarism checker.

4. Your report will be graded on the following criteria.
   - Computation
     (a) Does the report exhibit appropriate use of computational/software tools?
     (b) Does the report demonstrate complete understanding of the problem and related mathematical concepts?
     (c) Are all the required tasks fulfilled?
   - Communication
     (a) Is the report presented in a well-organized and logical manner, with proper formatting?
     (b) Are the statements well-structured and written in proper English?
     (c) Are the results and explanations clear and to the point?

C. **The Problem.** A local multi-industry company has just hired you as their first data analyst as they are transitioning from hard copies to digital data storage. They have plenty of employees doing the data entry of their company's numerical data which includes information like employee personal records, bank account numbers, product identification numbers etc. But they have no means of identifying the errors created by human or data transmission errors. Most errors that they have found to occur are single erasures or human typographical errors.

Having some experience with check digits while experimenting around with your NTU matriculation number when you were an undergraduate, you decide to create a program (MATLAB functions) that checks your company's data to correct single erasure errors and detect typographical errors.

The company also realises that some of their product's hexadecimal identification numbers i.e. MEID, ISAN are unable to detect certain types of errors under their respective check digit system. The company ask you to create a new check digit system which allows them to keep track of their product's identification numbers while having the same length but allowing it to have better error detection capabilities.

Section D and E will show the information on the various check digit systems which you should use for all of your implementation of the check digit system checkers.

C.1. *The Background.* A **check digit**[1] is a form of redundancy check used for error detection on codes, information data and identification numbers. A single check digit a_{n+1} is usually appended to a given sequence $a_1a_2\cdots a_n$ of information digits, but sometimes it can consist of more than one digit. Check digits are computed by an algorithm or equation from the other digits in the sequence input.

Some examples of one check digit systems:

- International Standard Book Number (ISBN 10)
- International Article Number (EAN-13)
- NTU Matriculation Number
- Credit Card Number
- Bank Account Number

Check digits is helpful in our lives since it can help detect common errors in data input, verify correct credit card number for online transactions and also verify correct bank account number for cheque deposits or fund transfers on ATMs.

C.2. *Common Types of Transmission Errors.* Most common human transmission errors (in order of complexity):

1. Single error: $\cdots a \cdots \to \cdots b \cdots$;
2. Adjacent transposition: $\cdots ab \cdots \to \cdots ba \cdots$;
3. Twin error: $\cdots aa \cdots \to \cdots bb \cdots$;
4. Jump transposition: $\cdots abc \cdots \to \cdots cba \cdots$;
5. Jump twin error: $\cdots aca \cdots \to \cdots bcb \cdots$;

The number beside each error will be the type of error in the following descriptions.

[1]For this project, the term 'digit' can mean either integers from 0-9 or letters from alphabet A-Z

C.3. *Hexadecimal Numbers.* Hexadecimal is a positional numeral system with a base of 16. Decimal values $0-15$ are represented as $0-9$, A, B, C, D, E, F in hexadecimal system. The following is an example for conversion from a hexadecimal to decimal number. E.g. $\mathrm{A9}_h = 10 \times 16^1 + 9 \times 16^0 = 169_d$. They are widely used by computer systems and digital electronics for its convenience of representing binary data since each hexadecimal digit represents four binary digits (bits).

C.4. *MEID and ISAN.*

- A Mobile Equipment IDentifier (MEID) is a globally unique 14 digit hexadecimal identification number for a physical piece of mobile station equipment. The check digit is calculated using a slight modification of the Luhn formula, and is usually not appended to the MEID.
- The International Standard Audiovisual Number (ISAN) is a numbering system that enables the unique and persistent identification of any audiovisual works. It consists of 16 hexadecimal digits, and the appended check digit is calculated over the the 16 ISAN digits according to a MOD 37,36 system, and is usually not appended to the ISAN.

Both of these widely used hexadecimal check digit systems, failed to detect all errors of types 1 - 5.

D. **The Check Digit Algorithms and Equations.**

D.1. *Check Digit Algorithm for NTU Matriculation Number.* Consider the 9 digits of a matriculation number. Let them be $d_1d_2d_3d_4d_5d_6d_7d_8d_9$, where d_9 is the check digit, and do the following calculations.

$$\sum = 5 \cdot d_2 + 4 \cdot d_3 + 7 \cdot d_4 + 8 \cdot d_5 + 9 \cdot d_6 + 2 \cdot d_7 + 3 \cdot d_8.$$

If the first digit $d_1 =$ G, let $n = \sum + 9$ otherwise if the first digit $d_1 =$ U, let $n = \sum + 6$. The check digit is then obtained from the following table where $m \equiv n \pmod{11}$.

m	0	1	2	3	4	5	6	7	8	9	10
d_9	L	K	J	H	G	F	E	D	C	B	A

D.2. *Check Digit Algorithm for NRIC Number.* Consider the 9 digits of a NRIC number. Let them be $d_1d_2d_3d_4d_5d_6d_7d_8d_9$, where d_9 is the check digit, and do the following calculations:

$$\sum = 2 \cdot d_2 + 7 \cdot d_3 + 6 \cdot d_4 + 5 \cdot d_5 + 4 \cdot d_6 + 3 \cdot d_7 + 2 \cdot d_8.$$

If $d_1 =$ T or G, let $n = \sum + 4$.

The check digit is then obtained from either of the following two tables, where $m \equiv n \pmod{11}$, depending on the first digit d_1.

If $d_1 =$ S or T, the following table is used.

m	0	1	2	3	4	5	6	7	8	9	10
d_9	J	Z	I	H	G	F	E	D	C	B	A

If $d_1 =$ F or G, the following table is used.

m	0	1	2	3	4	5	6	7	8	9	10
d_9	X	W	U	T	R	Q	P	N	M	L	K

D.3. *Check Digit Equation for International Standard Book Number (ISBN-10).* The ISBN-10 code $d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}$ uses the check equation.

$$\sum_{i=1}^{10} i \cdot d_i \equiv 0 \pmod{11},$$

where d_{10} is the check digit.

D.4. *Check Digit Equation for International Article Number (EAN-13).* Consider the 13 digits of a EAN-13 number. Let $d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}d_{11}d_{12}d_{13}$ be the digits of the numbers. The check digit is $d_{13} \equiv 10 - n \pmod{10}$, where

$$n = \sum_{k=1}^{6} (d_{2k-1} + 3 \cdot d_{2k}).$$

E. **Proposed Check Digit System for Hexadecimal Numbers.** The hexadecimal numbers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F can be represented as 4 bits by denoting $0 = (0,0,0,0), 1 = (0,0,0,1), \ldots,$ E $= (1,1,1,0)$ and F $= (1,1,1,1)$. Let $\mathbf{G}$ be the set of all these elements. The addition of two elements is bitwise over modulo 2, e.g. $(1,1,1,0) + (0,1,0,1) = (1,0,1,1)$.

Suppose that the hexadecimal information digits $d_1, \ldots, d_n$ and the check digit d_{n+1} are interpreted as the above representation. Then d_{n+1} can be determined by the check equation

$$P(d_1) + P^2(d_2) + \cdots + P^n(d_n) + P^{n+1}(d_{n+1}) = 0, \tag{1}$$

where P is a permutation of the set $\mathbf{G}$. Let us define the mapping $P : \mathbf{G} \to \mathbf{G}$ by

$$P((a,b,c,d)) = (a+b, c, d, a), \quad \text{where } a,b,c,d \in \{0,1\}. \tag{2}$$

The permutation P is an automorphism of $\mathbf{G}$ which contains a 15-cycle and fixes only the element 0. In cycle notation, it is

$$P = (1\,2\,4\,8\,9\,\mathrm{B}\,\mathrm{F}\,7\mathrm{E}\,5\,\mathrm{A}\,\mathrm{D}\,3\,6\,\mathrm{C})\,(0).$$

E.1. *Example 1.* Assume that the sequence of information digits is:

$$2\,4\,\mathrm{A}\,7\,\mathrm{F}\,6$$

Now $P(2) + P^2(4) + P^3(\mathrm{A}) + P^4(7) + P^5(\mathrm{F}) + P^6(6) = 4 + 9 + 6 + \mathrm{D} + \mathrm{D} + 9 = 4 + 6 = (0,1,0,0) + (0,1,1,0) = (0,0,1,0) = 2 = -2$ and as $P^7(5) = 2$, the check digit is 5 and the sequence ready for transmission is:

$$2\,4\,\mathrm{A}\,7\,\mathrm{F}\,6\,5$$

E.2. *Example 2.* Assume that the sequence of information digits with its check digit is:

$$\mathrm{B}\,\mathrm{E}\,2\,\mathrm{F}\,7\,1\,\mathrm{B}\,6\,6$$

$P(\mathrm{B}) + P^2(\mathrm{E}) + P^3(2) + P^4(\mathrm{F}) + P^5(7) + P^6(1) + P^7(\mathrm{B}) + P^8(6) + P^9(6)$
$= \mathrm{F} + \mathrm{A} + 9 + \mathrm{A} + 3 + \mathrm{F} + 3 + \mathrm{F} + 7$
$= 9 + \mathrm{F} + 7 = (1,0,0,1) + (1,1,1,1) + (0,1,1,1)$
$= (0,0,0,1) = 1 \neq 0$
and we conclude that an error has occurred.

E.3. *Generalization of Permutation P into Matrix Form.* A matrix representation of P is

$$P = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3}$$

where P is a 4×4 matrix. The equation (2) can now be written as

$$P((a,b,c,d)) = (a,b,c,d) \cdot P^2. \tag{4}$$

As a direct result, the check equation (1) can be written as

$$d_1 \cdot P + d_2 \cdot P^2 + \cdots + d_n \cdot P^n + d_{n+1} \cdot P^{n+1} = 0. \tag{5}$$

This hexadecimal check digit system using this specifi P is able to detect all errors of types $1 - 5$.

[2] Note that each entry is taken over modulo 2

F. **Your tasks.** Your specific tasks are as follows.

1. NTU Matriculation Number Checker
   Your first task is to design a NTU matriculation number checker, which allows you to take any input string and give you its correct output.
   Create a *general purpose* function `MATRIC` that performs checking for errors and finding missing digits of the NTU matriculation number.
   The function should have 2 inputs, the matriculation number and a boolean value, indicating TRUE for checking of errors and FALSE for finding missing digits. The function should also allow input with multiple missing digits (including the check digit), which should be noted as _ at the missing positions. The function should return an output indicating an error if input is not a valid matriculation number (e.g. 'S1234567A' is invalid since it should start with U or G).
   Use your function to complete the following task.
   (i) Return an output indicating an error if input is not a valid matriculation number,
   (ii) Input the matriculation number 'U1401234_' and output the correct full matriculation number with its check digit.
   (iii) Input the matriculation number 'U1510936G' to check if there are any errors. If there are, output any possible corrected matriculation numbers.
   (iv) Suppose there was an erasure of one digit on the matriculation numbers 'U1240_60L' and 'G1301_81E'. Output the correct matriculation number for both of these cases.
   (v) Now suppose there are a few missing digits on the matriculation numbers 'U140_2_3A' and 'G150_931_'. Output all possible matriculation numbers with the given digits.

2. NRIC Number Checker
   From your experience with the NTU matriculation number checker in Task 1, create a *general purpose* function `NRIC` to check for similar errors in the NRIC number. The exception is your function does not have to output all possible correct NRIC numbers if there is an error. Your function should have 2 inputs, the NRIC number and a boolean value, indicating TRUE for testing of errors and FALSE for finding missing digits. The function should also allow input with multiple missing digits (including the check digit), which should be noted as _ at the missing positions. The function should return an output indicating an error if input is not a valid NRIC number (e.g. 'S1234567X' is invalid since NRIC number starting with S cannot end with X).
   Use your function to complete the following task.
   (i) Input the NRIC number 'S9523142_' and output the correct NRIC number with its check digit.
   (ii) Input the NRIC number 'T0123456A' to check if there are any errors.
   (iii) Suppose there was an erasure on the NRIC numbers 'F219321_X' and 'T1502_01G'. Output the correct NRIC number for both of these cases.
   (iv) Now suppose there are a few missing information on the NRIC numbers 'S75_28_1G' and 'G840_253_'. Output all possible NRIC numbers with the given digits.

3. <u>ISBN-10 Checker</u>
   Create a *general purpose* function `ISBN` that allows you to check for errors for ISBN-10 and find *single* missing digits. The function should allow input with at most one missing digit, which should be noted with _ at the missing position. The function should return an output indicating an error if input is not a valid ISBN number.
   Use your function to complete the following task.
   (i) Input the ISBN '073342609_' and output the correct ISBN number with its check digit.
   (ii) Input the ISBN '0901690546' and '8515412937' to test if there are any errors.
   (iii) Suppose there was an erasure on the ISBN '_533613407'. Output the correct ISBN.
   The ISBN-10 check digit system is able to detect almost all errors from type $1-5$ for all ISBN. The exception is a type 3 error (twin error) occurring at a particular position in the ISBN. Find this position and explain why it is unable it by using the function `ISBN` on 2 ISBN with that error.
4. <u>EAN-13 Checker</u>
   Create a *general purpose* function `EAN` that allows you to check for errors for EAN-13. The function should allow input with at most one missing digit, which should be noted with _ at the missing position. The function should return an output indicating an error if input is not a valid EAN number.
   Use your function to complete the following task.
   (i) Input the EAN '501401615082_' and output the correct EAN number with its check digit.
   (ii) Input the EAN '9790260000438' and '9782123957123' to check if there are any errors.
   (iii) Suppose there was an erasure on the EAN '659426_856417'. Output the correct EAN.
   Will EAN-13 check digit system be able to detect all errors from type $1-5$? Explain your answer with evidence by using certain outputs from the function `EAN`.
5. <u>Hexadecimal Check Digit System Checker</u>
   By referring to the check digit system proposed in Section E, create a *general purpose* function `HEXA` that allows you to check for errors and find check digits for any types of hexadecimal identification numbers, e.g. MEID and ISAN.
   Your function will require 2 inputs, the hexadecimal number and a boolean value, TRUE for checking for errors, and FALSE for finding check digits. Your hexadecimal input should contain only the identification numbers itself when finding check digits, but contain both the identification number and check digit appended at the end when checking for errors.
   Use your function to complete the following task.
   (i) Input the 16 digit hexadecimal identification number '19A93E956C81B0F2' and return the check digit as output.
   (ii) Input the hexadecimal number (inclusive of check digit) '523AF23EB1C29' and 'ABCDEF0123456789' to check if there are any errors. Return an output indicating if there is an error or not.

(Important note: All input of identification numbers for all tasks should be given as a string of characters. E.g. for input of matriculation number, 'U1234567A'.)