Create function(s) that create a substitution cipher encoder and decoder and a transposition cipher encoder and decoder, plus functions that demonstrate these ciphers. Some useful MATLAB functions: `randperm`, `double`, `char`, and `reshape`. All files created must contain the first four letters of your last name.

Substitution Cipher

The encoder function will create a basic substitution cipher on an "extended alphabet". The encoder function will have two inputs: the first is string (entered in single quotes) that is a filename for a text file in which the text may include numbers, characters, and spaces. The second input will be a flag (either 1 or 0) as to whether the messages are displayed.

You will use the command `fileread(filename)` to open the file and store the text as one long vector of strings. Any characters that have an ASCII code of 32-126 are allowed: see http://www.asciitable.com/. (ASCII stands for "American Standard Code for Information Interchange.") The encoder function will then encode the text using a random substitution cipher. The function will return a string that is the encoded text along with a matrix of characters that will be the key. The top row of the matrix will be the original alphabet (represented by the ASCII code), and the second row will be the substituted alphabet (represented by the ASCII code). Also, if the flag is true for displaying the messages, the encoder function will display both the original and encoded message to the screens. The display will not be as one long string; instead, the messages will be formatted to be split across several lines so that no more then 50 characters are on one line. If the flag is false, no output is displayed to the screen (unless the user chooses to see the output of the function).

The decoder function will have three inputs; the key (which would be the matrix that the encoder function gives you), the filename of the encoded text, and the flag for displaying the messages to the screen. Using the key, the decoder function will return a string of text that is the decoded message. If the flag is true for displaying the messages, then both the encoded message and decoded message are displayed to the screen as in the encoder function. If the flag is false, no output is displayed to the screen (unless the user chooses to see the output of the function).

IMPORTANT: do not choose a specific substitution cipher - have MATLAB randomly choose it so that each time you run the encoding function, you may get a different encoded message (thus the need for the key to decode it).

To demonstrate the above functions, create another function that will create tables that show the substitution cipher. It will as input take the "key" and create tables of the character and what that character looks like in the encoded message. Have one table be the characters (may split up into multiple tables for visual purposes), another table the numbers, and one or two tables the letters (split between upper and lower? you be the judge). Thus your output *may* look like this (example cipher shown):

```
 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
-----------------------------------------------------
 D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ]


 a b c d e f g h i j k l m n o p q r s t u v w x y z
-----------------------------------------------------
 d e f g h i j k l m n o p q r s t u v w x y z { | }
```

```
 0 1 2 3 4 5 6 7 8 9
--------------------
 3 4 5 6 7 8 9 : ; <
```

```
   ! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~
-------------------------------------------------------------------
 # $ % & ' ( ) * + , - . / 0 1 2 = > ? @ A B C ^ _ ` a b c ~   ! "
```

This would signify that for every "A" in the original message, it is encoded as "D", every space becomes the character "#", etc. Make these tables be visually pleasing in the output window (using `fprintf`, etc.)

Columnar Transposition Cipher

For the columnar transposition cipher, the encoding function has as input filename of the original text (similar as above: characters, spaces, numbers, etc. are allowed), the number of columns to use, and a flag for displaying the messages (similar as above). The number of rows in the "matrix" is determined by the number of characters in the given text; if there are not enough characters to completely fill the matrix, dummy (random) characters are put at the end of the text to complete the matrix. The output will be the encoded text and a key which is a vector that tells the order of the permuted columns. The decoder function will have as input the filename that contains the encoded text, the key (vector), and a flag for displaying the messages. The output is the decoded message.

Another function will demonstrate the columnar transposition. It will take as input the original message and the key. The function will display the original message, the matrix of characters, the permuted matrix of characters, and the encoded message. These displays should be formatted such that if the messages are long, then no more than 50 characters are displayed on one line.

Your report should include a basic history of these types of ciphers, examples, pros and cons to using these types of ciphers, and other variations of the ciphers.