

**ENCMP 100 – Computer Programming for Engineers**  
**Assignment #2**

**Due: Monday, Oct. 14, 2013 6:00pm MST**

### Objective

This assignment is designed to provide you with practice using selection statements to solve an encoding problem. There are many applications for encoding and decoding information in engineering contexts. For example, in the area of Computer Engineering there is often a need to transmit information from one place to another, e.g. the Internet.

In some situations, the information must be converted into a form suitable for transmission from the source to the destination. The encoded information must then be decoded at the destination before it can be used. There are several ways to encode and decode information, e.g., some are designed for secure data transmission, while others decrease the size of the data to be stored or transmitted (compression).

### Marking Scheme

This assignment is worth 3% of your final mark. You will get a total of 50 points for completing the following:

TASK	POINTS
Successful testing of Rules	35
Quality of code	15
TOTAL	50

#### Points for Quality of Code

- Complete file header (see Hints for an example) – 3 point
- Design (appropriate use and naming of variables) – 5 points
- Comments in the code – 5 points
- Layout (indentation/spacing) – 2 point

### Submission

- Filename to be used for this assignment Assign2\_<UofA\_ID\_Number>.m  
Ex. U of A ID Number: 1234567890 filename for assignment #2 is  
Assign2\_1234567890.m
- Submit only your **.m** file under Assignment 2 in your eClass/Moodle account.
- The assignment is due on Monday, Oct. 14 at 6:00 pm MST

## Background

Your friend Maverick recently landed a lucrative engineering co-op work term in an oil-rich country. Meanwhile, you chose to make a modest salary as a co-op intern with the [Canadian Security Intelligence Service](#) (CSIS). A week after arriving in the foreign country, Maverick is kidnapped by rebels opposed to oil exploration. He is held captive in a small village within rebel territory. Luckily, a group of disenchanted rebels are willing to smuggle Maverick to one of seven rendezvous points:

- the village bridge
- the village library
- river crossing
- nearby airport
- bus terminal
- hospital
- at St. Pete's Church

where an allied helicopter will pick them up and fly them to safety. For the plan to work, the friendly rebels need to know two pieces of information:

- the day of the rendezvous
- the rendezvous point

CSIS has devised a seven-digit code to carry this information to the friendly rebels. When the time is ripe, CSIS will insert a coded message into the online newspaper of the oil-rich country, as a number on the front page. The rebels have Internet access via satellite links and follow the news. While it is helpful (as a decoy strategy) that the newspaper has other numbers on the front page, e.g. in real adverts, it is vital that the rebel insiders can detect and decode the secret message. CSIS has found a way to smuggle a decoder to the friendly rebels. You are asked to program the decoder in MATLAB. The safety of your friend Maverick depends on your ability to program the decoder correctly.

## Details

The four rules for detecting and decoding the code are as follows:

1. A valid code must be a 7-digit number.
  - e.g. 1234567, message MAY be valid
  - e.g. 12345, message invalid
  - e.g. 123456, message invalid

Assume that the input data contains only digits, with a non-zero first digit.

2. The code must pass the odd-even "truth" test.

- If the sum of the digits is odd, the message is invalid.  
e.g. 2222223, sum = 15, message invalid
- If the sum of the digits is even, the message MAY be valid.  
e.g. 7272211, sum = 22, message may be valid

3. To determine the rescue day, multiply the 1<sup>st</sup> digit by the 2<sup>nd</sup> digit, then subtract the 3<sup>rd</sup> digit. The answer indicates the rescue day, as follows:

1 = Monday    2 = Tuesday    3 = Wednesday  
4 = Thursday    5 = Friday    6 = Saturday    7 = Sunday

Any other number indicates that the message is invalid.

Examples: 3249621     $3 \times 2 - 4 = 2$     Rescue is on Tuesday  
7245421     $7 \times 2 - 4 = 10$     Message is invalid.

4. The last four digits determine the rendezvous point.

- If the sum of the fourth and fifth digits is greater than the sum of the sixth and seventh digits, the rendezvous number is the sum of the fourth and fifth digits minus the sum of the sixth and seventh digits;
- If the sum of the fourth and fifth digits is less than the sum of the sixth and seventh digit, the rendezvous number is the sum of the sixth and seventh digits minus the sum of the fourth and fifth digits;
- If the sum of the fourth and fifth digit is equal to the sum of the sixth and seventh, the message is invalid.

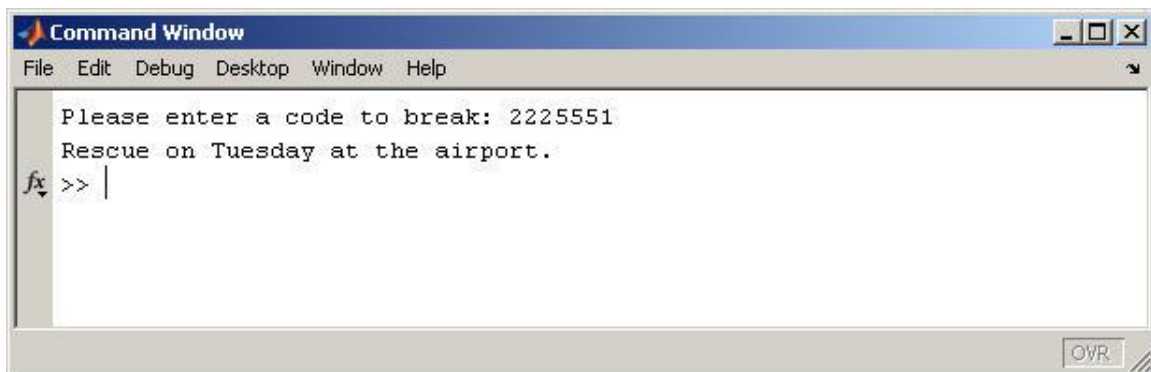
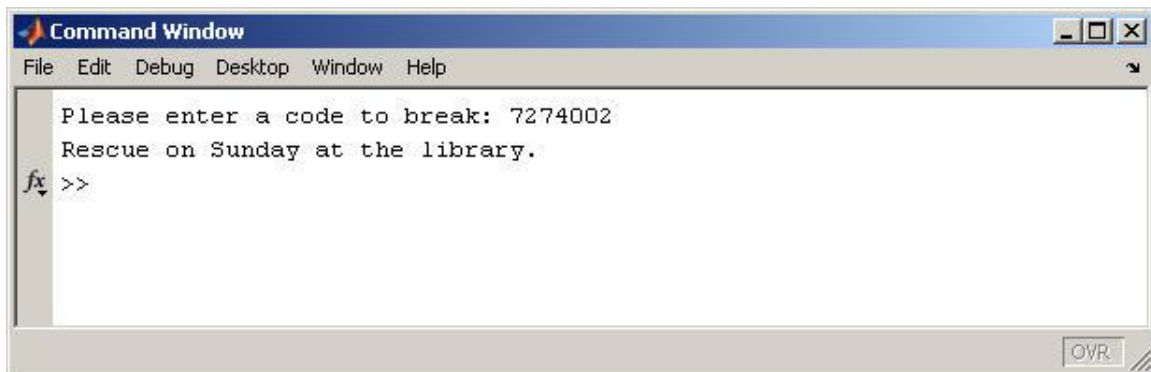
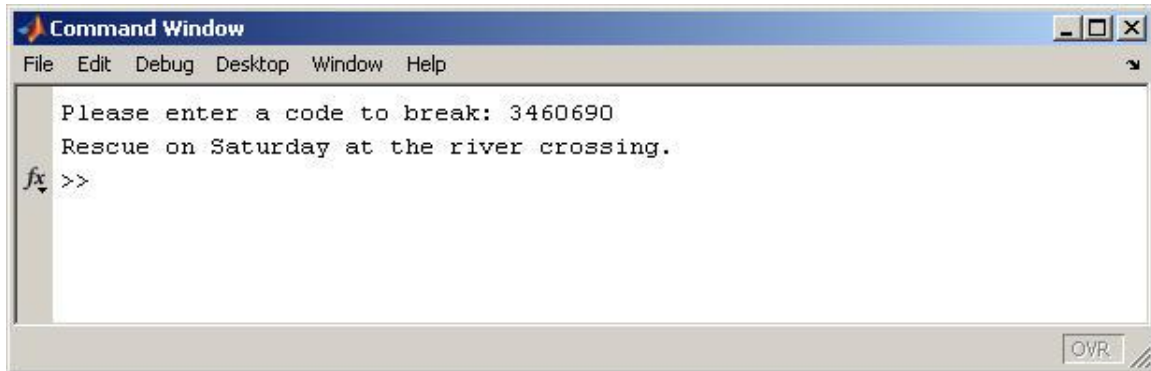
Rendezvous Number	Rendezvous Point
1	bridge
2	library
3	river crossing
4	airport
5	bus terminal
6	hospital
7	St. Petes Church

If the result is one of the following rendezvous numbers, the message is valid:

Any other rendezvous number means that the message is invalid.

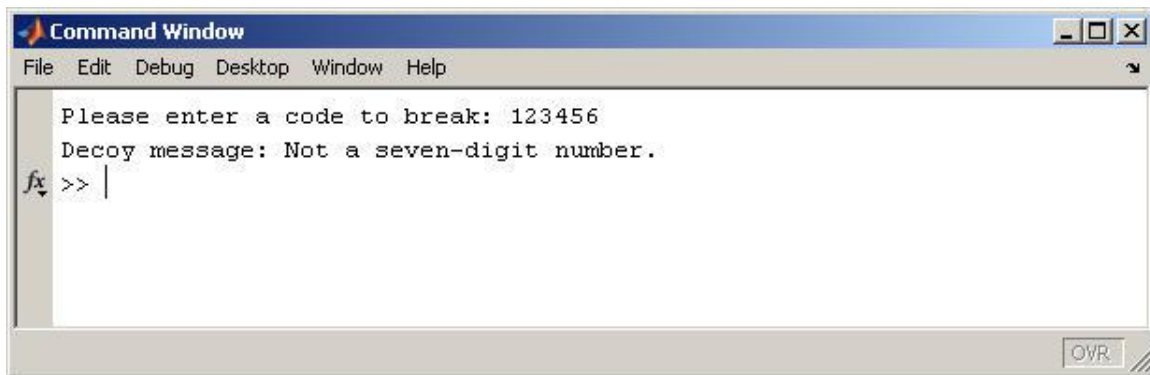
## Output

Your output should look like the example screenshots below. Here are 3 valid messages:



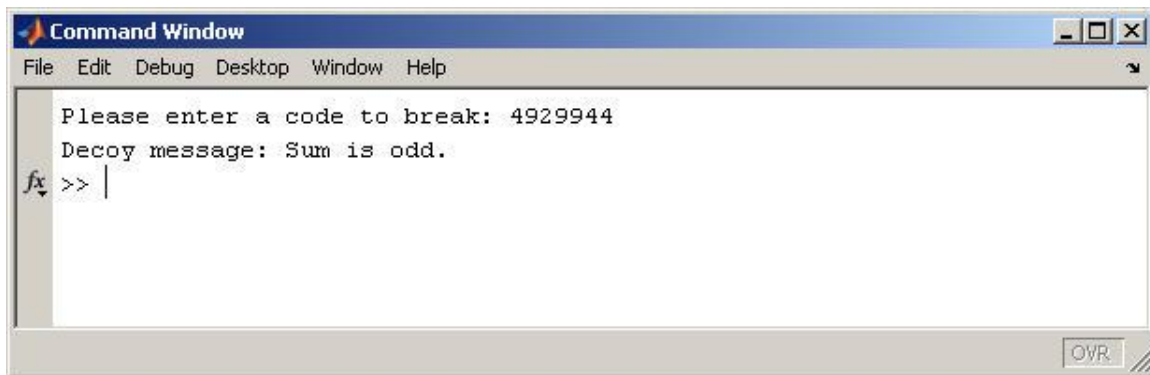
Your output for invalid messages should indicate only the first reason encountered for rejecting the message. The reasons are one of the following, in this order:

- Error Message: Not a seven-digit number.
- Error Message: Sum is odd.
- Error Message: Invalid rescue day.
- Error Message: Invalid rendezvous point.



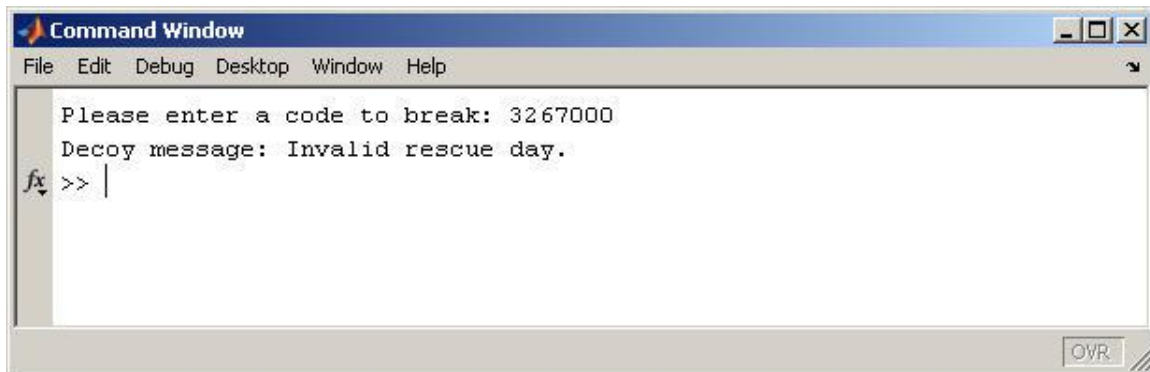
A screenshot of a 'Command Window' interface. The title bar is blue with the text 'Command Window' and standard window controls. The menu bar includes 'File', 'Edit', 'Debug', 'Desktop', 'Window', and 'Help'. The main text area contains the prompt 'Please enter a code to break: 123456' followed by the message 'Decoy message: Not a seven-digit number.' Below this, the prompt 'fx >> |' is shown with a cursor. At the bottom right, there is a button labeled 'OVR'.

```
Command Window
File Edit Debug Desktop Window Help
Please enter a code to break: 123456
Decoy message: Not a seven-digit number.
fx >> |
OVR
```



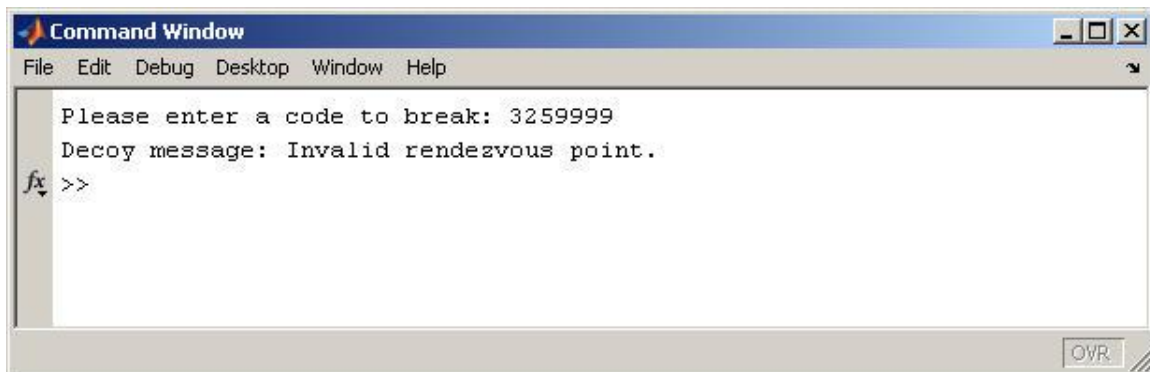
A screenshot of a 'Command Window' interface. The title bar is blue with the text 'Command Window' and standard window controls. The menu bar includes 'File', 'Edit', 'Debug', 'Desktop', 'Window', and 'Help'. The main text area contains the prompt 'Please enter a code to break: 4929944' followed by the message 'Decoy message: Sum is odd.' Below this, the prompt 'fx >> |' is shown with a cursor. At the bottom right, there is a button labeled 'OVR'.

```
Command Window
File Edit Debug Desktop Window Help
Please enter a code to break: 4929944
Decoy message: Sum is odd.
fx >> |
OVR
```



A screenshot of a 'Command Window' interface. The title bar is blue with the text 'Command Window' and standard window controls. The menu bar includes 'File', 'Edit', 'Debug', 'Desktop', 'Window', and 'Help'. The main text area contains the prompt 'Please enter a code to break: 3267000' followed by the message 'Decoy message: Invalid rescue day.' Below this, the prompt 'fx >> |' is shown with a cursor. At the bottom right, there is a button labeled 'OVR'.

```
Command Window
File Edit Debug Desktop Window Help
Please enter a code to break: 3267000
Decoy message: Invalid rescue day.
fx >> |
OVR
```



A screenshot of a 'Command Window' interface. The title bar is blue with the text 'Command Window' and standard window controls. The menu bar includes 'File', 'Edit', 'Debug', 'Desktop', 'Window', and 'Help'. The main text area contains the prompt 'Please enter a code to break: 3259999' followed by the message 'Decoy message: Invalid rendezvous point.' Below this, the prompt 'fx >>' is shown with a cursor. At the bottom right, there is a button labeled 'OVR'.

```
Command Window
File Edit Debug Desktop Window Help
Please enter a code to break: 3259999
Decoy message: Invalid rendezvous point.
fx >>
OVR
```

## Code Requirements

You are required to use **two switch statements** in this assignment, one to determine the rescue day and another to determine the rendezvous point. Failure to use two switch statements in this assignment will result in a mark of zero.

## Hints

1. Before you even begin to write out any MATLAB code, sit down and sketch out your program design in abbreviated English. This kind of high-level program design is called pseudocode. Please refer to page 59 of the combined notes pdf or slide 55 of section 05\_Selection. It would start off something like this:

```
Get secret code from user as a string
Get an array of the individual digits
If code not 7 digits
    Output error message
Else
    Get sum of digits
    If sum is odd
        Output error message
    Else
        Get rescue day
        Etc, ...
```

Notice how the logic naturally translates into an `if-else` chain. The `if` clauses in the chain contain messages for invalid codes. After such a message is output, the control flow breaks out of the entire chain. You end up reaching the final `else` clause in the chain only if the code is valid. It is there that the valid message is composed and output.

For the complete pseudo code for this assignment, please attend your assigned lab section.

2. To help you read in a secret code from the user as a string, check out the lectures notes page 30, section 03\_BasicsPart1, slide 54.

3. When converting an array of characters (string) to an array of numbers, you need to subtract the ASCII value of zero from each character. For example, if your string is called `code_str` and your number array is called `digits`, you can do the following:

```
digits = code_str - '0';
```

4. Remember to use the built-in MATLAB functions. For example, to check if the string contains seven digits, you might find the **`length`** function helpful. To find the sum of the digits, you might find the **`sum`** function helpful. To check if the sum of digits is odd, you may find the **`mod`** function helpful.

5. Build up your program design in stages. After you have completed the code for Rule 1, compile and test with a variety of codes that satisfy and fail to satisfy the rule. Then go on to write the code for the next rule, etc.

**6. Please remember to include your header at the top of your .m file.**

[illegible]