A. Submission Instructions:

- Submit your solutions via eLearning.
- Please include **ALL** the solutions in a **single** PDF/Doc/PS/Image file
  - The filename should follow the pattern "HW#_FirstnameLastname.FileExtension" (Example: HW3_ClaireUnderwood.pdf)
  - **Penalty of 5 points** if not followed
- Late Submission Penalty:
  - up to 2 hours late — 10% deduction
  - 2 - 4 hours late — 20% deduction
  - 4 - 12 hours late — 35% deduction
  - 12 - 24 hours late — 50% deduction
  - 24 - 48 hours late — 75% deduction
  - more than 48 hours late — 100% deduction (zero credit)

B. Problems:

Please note that **ONLY** operators presented in the Lectures can be used to answer Regex questions in the homeworks and exams. You **CANNOT** use lookahead operator, lookbehind operator, etc.

1. Regular Expressions (30 points)

Write regular expressions for the following. Your Regular Expression should find the largest matching string. By "word", I mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth.

1. the set of all alphabetic strings

   Examples:
   > why that is gr8!
   > No, it is not gr8 at all!

2. the set of all alphabetic words

   Examples:
   > why that is gr8!
   > No, it is not gr8 at all!

3. the set of all lower case alphabetic strings ending in a b

    Examples:

    > Many programming languages provide regex capabilities, built-in,
    > or via libraries.
    > Please use tab.

4. the set of all lower case alphabetic words ending in a b

    Examples:

    > Many programming languages provide regex capabilities, built-in,
    > or via libraries.
    > Please use tab.

5. the set of all strings from the alphabet {"a", "b"} such that each "a" is
    immediately preceded by and immediately followed by at least one "b"

    Examples:

    > The use of babble helps.

    > Tab is not bob's bbabled bass.

6. the set of all words from the alphabet {"a", "b"} such that each "a" is
    immediately preceded by and immediately followed by at least one "b"

    Examples:

    > The use of babb helps.

    > Tab is not bb in bob's bbabled bab.

7. the set of all strings from the alphabet {"a", "b"} that form the pattern $a^n b^m$
    where (n+m) is even; n>=0, m>=0, and (n+m)>0

    Examples:

    > The use of baabble helps.

    > Tab is not a bb in aa bob's baaabbbled bass.

2. Write a single regular expression for identifying social security numbers in text. (30 points)

The social security numbers consists of:

- 9 digits
- must be preceded by one or more spaces or beginning of line
- must be followed by one or more spaces or ends of line

In addition there are certain restrictions:

- first three digits cannot be all zeros
- last four digits cannot be all zeros
- nine digits can all be next to each other

    or

    there can be a hyphen between:

    - third and fourth digit, and
    - fifth and sixth digit

The following are well formed social security numbers: 123456789, 123-45-6789.

The following are ill-formed social security numbers: 000-23-4567, 123-45-0000.

There is no valid social security number on the following line:

12345678910 is a big number, 345-678-910 is a lotto number and 3333333334 is a 10 digit number.

3. Telephone Number (40 points)

Create a single regular expression and a FSA for a telephone number that comply with the following patterns (surrounded by word boundaries):

    i. +(country_code)-(area_code)-(prefix)-(line_number)
    ii. +country_code-area_code-prefix-line_number

"country_code" is a two digit string value except "00"

"area_code" and "prefix" are a three digit string value except "000"

"line_number" is a four digit string value