CSE102 – Homework 6
Submission Deadline: November 28, 2016, 11:55 pm

Email spam, also known as junk email, is a type of unsolicited (uninvited) message that is sent by email. It can contain disguised links that may lead to phishing or malware web sites. It may also include malware or other executable file attachments. There are various methods to detect email spam. In this assignment we will focus on a basic technique called keyword filtering, checking the presence of certain words either separately or in groups. Email messages contain two major sections, the message header and the message body. The message header contains different fields, such as To, CC, Subject, Date, etc. The message body contains the email message text. In this assignment we will only read the Subject field and the message body as plain text.

Keyword Filtering:
First we are going to extract the bag of words, i.e, sequences of tokens separated by spaces, or punctuation marks. Presence of a certain word(s) (keyword/token) in the message is considered a feature of the message. We will also assign a weight to each token which is the number of occurrences of the token in the message. This reflects the importance of the token in the message.

After extracting all these features, we will have to decide if the message is a spam or a legal email (we will do this in the next assignment). You will be provided with a file that contains the keywords/tokens and their respective frequencies (number of occurrences in the message) to classify an email message either spam or legal. You will read this file and build your data structure accordingly to classify the email message. The format of this file (token.txt) is as follows:

```
$$$ = 1
Cents on the dollar = 2
Fast cash = 2
Unsecured debt = 2
Avoid bankruptcy = 2
Additional Income = 3
Customer Alert = 3
Work at home = 2
Incredible deal = 3
For just $XXX = 1
http://e-dlogs.rta.mi.th = 1
http://servicing.capitalone-iv.com = 1
```

You will also be provided with a file (emails.txt) containing the email messages. Each email will be enclosed in specific tags, highlighting different fields, as follows:
(ref: https://itservices.uchicago.edu/page/examples-email-scams)

<email>
<Subject> Get your tax refund now </Subject>
<Body> After the last annual calculations of your account activity we have determined that you are eligible to receive a tax refund of $479.30. Please submit the tax refund request and allow us 2-6 days in order to process it.

A refund can be delayed for a variety of reasons. For example submitting invalid records or applying after the deadline.

To access the form for your tax refund, please click here
(http://e-dlogs.rta.mi.th:84/www.irs.gov/)

Note: Deliberate wrong inputs will be prosecuted by law.

Regards,
Internal Revenue Service</Body>
</email>

<email>
<Subject>Customer Alert</Subject>

<Body>Dear Capital One Cardholder,

During our regularly scheduled account maintenance and verification procedures, we have detected a slight error regarding your Capital One Card(s).

This might be due to one of the following reasons:

1. A recent change in your personal information (i.e. address changing) 2. Submitting invalid information during the initial sign up process.
4. Multiple failed logins in your account.
3. An inability to accurately verify your selected option of payment due to an internal error within our system.

Please update and verify your information by clicking the following link:

http://servicing.capitalone-iv.com/c1/login.aspx

Note: You must verify your information before you can continue using your card.

Thank you,
Capital One.</Body>
</email>

REQUIREMENTS:

1. The program should read both files (the token file and the file containing emails), parse the files, check for errors and print the line number where error detected.
Following errors to be checked:
The frequencies listed in the token file are numbers . e.g: Fast cash = two is not valid.
There is an equal (=) sign before the frequency in the token file to indicate the number. e.g: Fast cash # 2 is not valid.
The email, subject and body tags are present in the email file and are valid, i.e, the opening and closing tags there. e.g: <body> blah blah . . . <body> is not valid, and so on for other tags.

2. After checking the errors the program should print the results of parsing the token file (n number of tokens) in a table as follows:
Token file <filename>. Total tokens read n:

```
-----------------------------------------------
|  #  |    token              | Frequency |
-----------------------------------------------
|   1 | Fast cash            |              2 |
|   2 | Unsecured debt  |              2 |
|   3 | . . . . . .                |               x |
|   n | . . . . . .                |               x |
-----------------------------------------------
```

The dots . . . . in the table means that you need to fill them up with the parsing data from the token file.

3. After checking the errors the program should print the results of parsing the email file (n number of emails) in a table as follows:
Email file <filename>. Total emails read n:

```
-----------------------------------------------------------------------------------------------
|  #  |    subject       |      body                                                        |
-----------------------------------------------------------------------------------------------
|   1 | Get your tax   | After the last annual calculations of your account |
|      | refund now     | activity we have determined that you are eligible  |
|      |                      | to receive a tax refund of $479.30 . . . . . . .          |
|   2 | . . . . . .          | . . . . . . . . . . . . . . . . . . . . . . . . .                           |
|   n | . . . . . .          | . . . . . . . . . . . . . . . . . . . . . . . . .                           |
-----------------------------------------------------------------------------------------------
```

The dots . . . . in the table means that you need to fill them up with the parsing data from the email file.

NOTES:
- You can use the above files token.txt and emails.txt as sample files to test your program.
- Requirement 1 is worth 40 points and the other two requirements are each worth 30 points.

RULES:
1. Obey honor code principles.
2. Read your homework carefully and follow the directives about the I/O format (data file names, file formats, etc.) and submission format strictly. Violating any of these directives will be penalized.
3. Obey coding convention.
4. Your submission should include the following file and NOTHING MORE (no data files, object files, etc):

   HW06_<Firstname>_<Lastname>_<student number>_ praser.c

   Do NOT compress the files you submit.

5. Do not use non-English characters in any part of your homework (in body, file name, etc.).
6. Deliver the printout of your work until the last submission date.