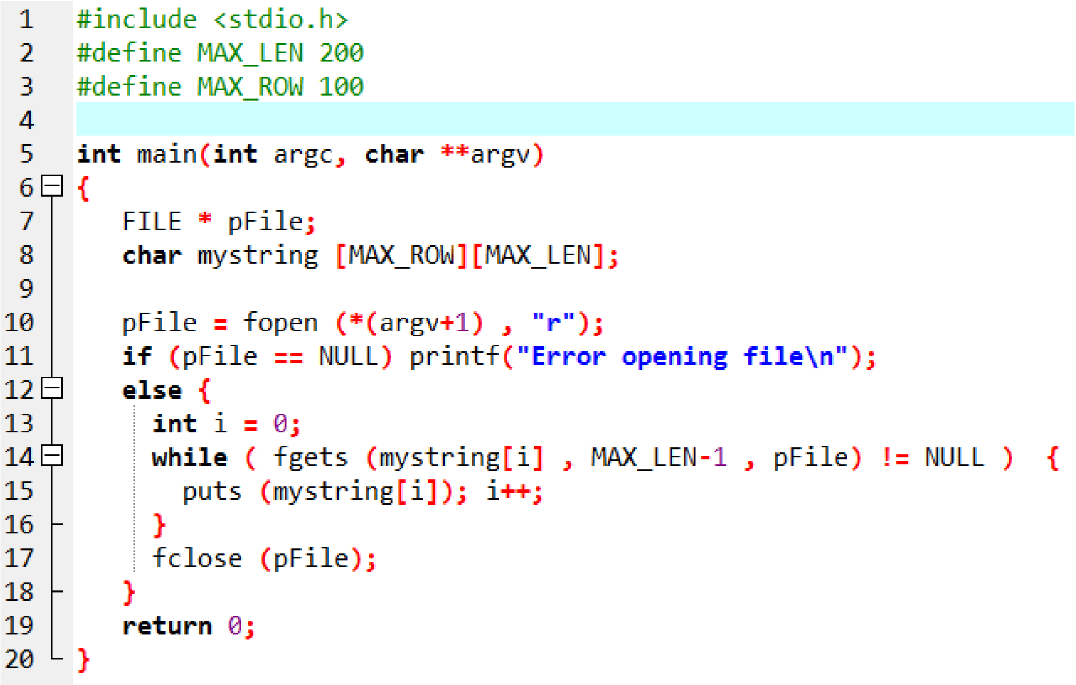
Practice 9 (2015/12/1)
SPAM email checker

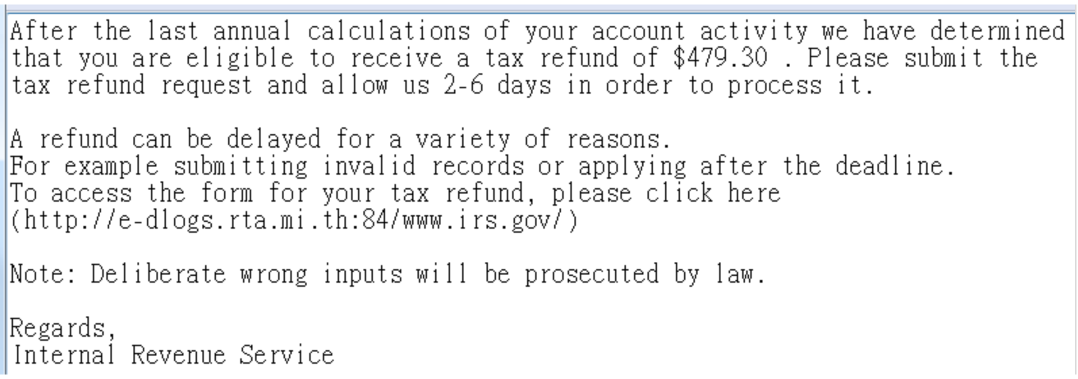
1. File IO

A. *fopen, fclose, fgets*

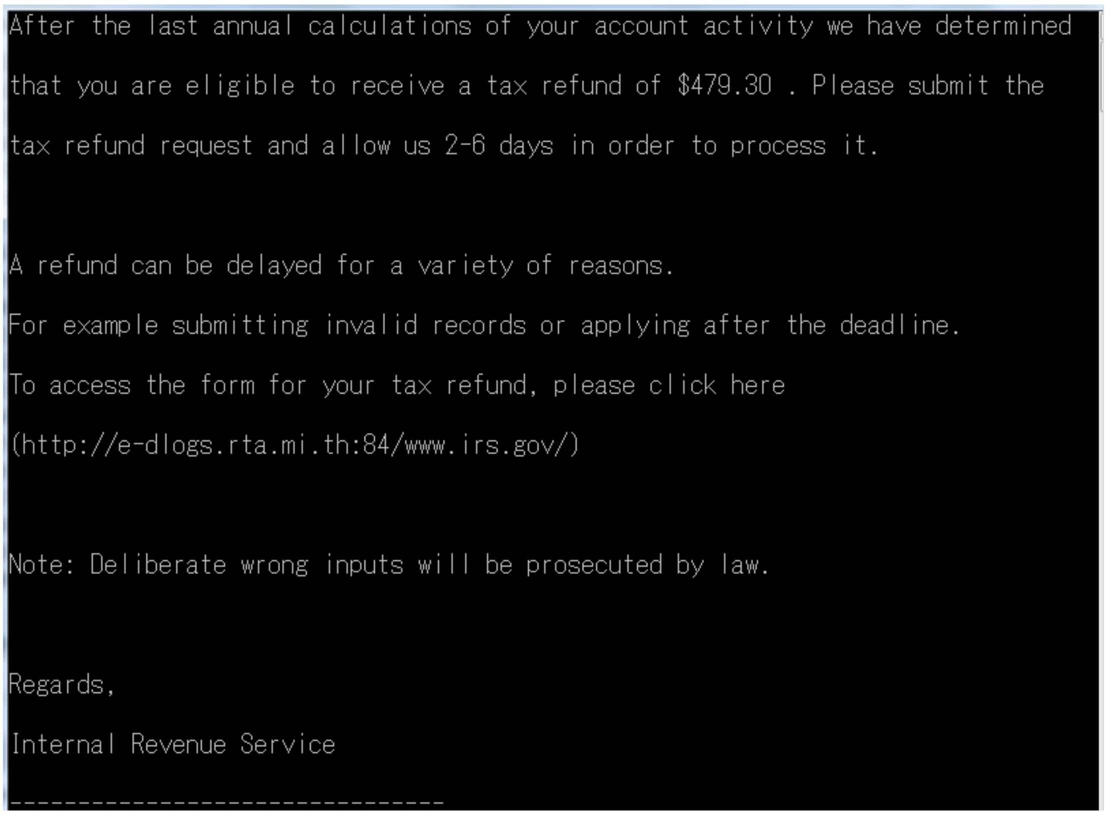
B. Example



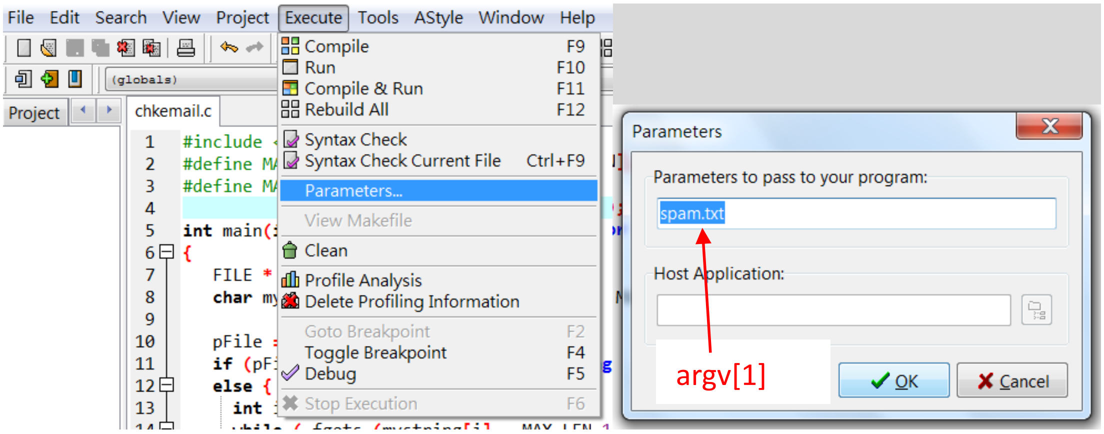
```c
#include <stdio.h>
#define MAX_LEN 200
#define MAX_ROW 100

int main(int argc, char **argv)
{
    FILE * pFile;
    char mystring [MAX_ROW][MAX_LEN];

    pFile = fopen (*(argv+1) , "r");
    if (pFile == NULL) printf("Error opening file\n");
    else {
      int i = 0;
      while ( fgets (mystring[i] , MAX_LEN-1 , pFile) != NULL )  {
        puts (mystring[i]); i++;
      }
      fclose (pFile);
    }
    return 0;
}
```



```
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
```



```
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

--------------------------------
```



```
#include <
#define MA
#define MA

int main(i
{
    FILE *
    char my

    pFile =
    if (pF
    else {
      int i
```

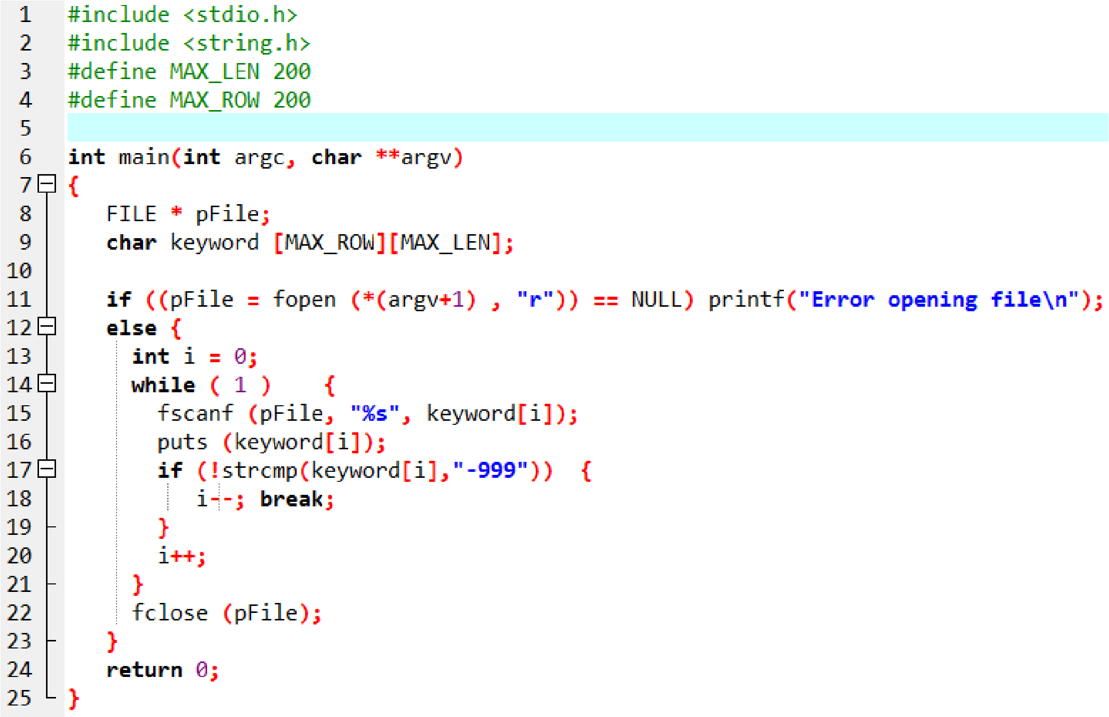
Parameters to pass to your program: spam.txt

2. Problem

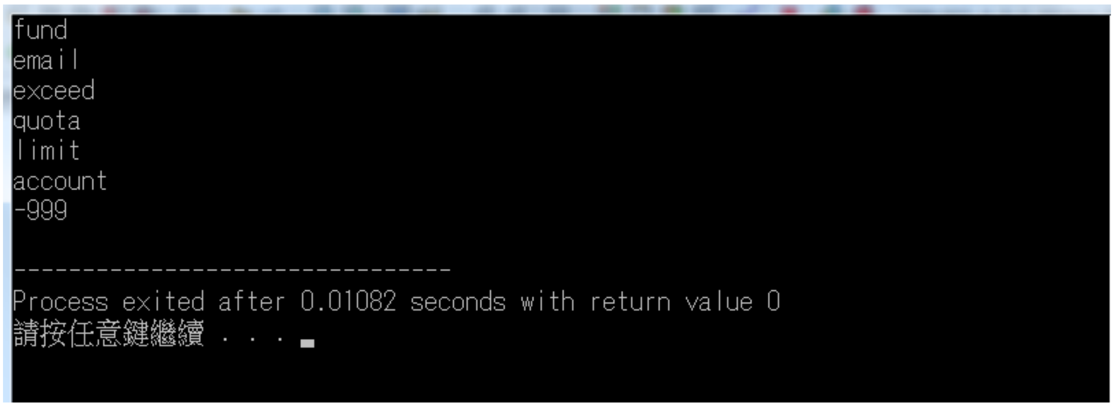
SPAM emails consume many network resources and cost U.S. organization billions of dollars a year in spam-prevention software, equipment, network resources, bandwidth and lost productivity. Write a program to read an email and save its content into a 2-D character array (one line from the first character to the newline in a row). **Initially you are given several spam email examples (files), a set of keywords for each type of spam emails (each type in a file), and a set of weights for each spam keyword.** For each spam keyword, find out the number of occurrences in an email. Then you can calculate the total cost for an email and determine if this is a spam email. Here is a simple and straightforward way to define the weight for each keyword and compute the total weight for an email. For each keyword's weight, you can just simply use the appearing rate of a keyword among all keywords as its weight. For instance, you have eleven keywords and all keywords appear 100 times in total in your spam email examples. For keyword A, it only appears 3 times, then A's weight is 0.03. As for the way to compute the spam weight for an email is to use this formula:

$$sw_e = \sum_{k \in KW} w_k \times \frac{a_k}{N_w}$$

Where sw_e is the total spam weight for email *e*, *KW* is the set of keywords, w_k is the weight of keyword *k*, N_w is **the total number of words of the email** under analysis, and a_k is **the number of occurrences of keyword *k* in this email**. As sw_e is higher than a threshold value, this email is regarded a spam email.



```
#include <stdio.h>
#include <string.h>
#define MAX_LEN 200
#define MAX_ROW 200

int main(int argc, char **argv)
{
   FILE * pFile;
   char keyword [MAX_ROW][MAX_LEN];

   if ((pFile = fopen (*(argv+1) , "r")) == NULL) printf("Error opening file\n");
   else {
     int i = 0;
     while ( 1 )    {
       fscanf (pFile, "%s", keyword[i]);
       puts (keyword[i]);
       if (!strcmp(keyword[i],"-999"))  {
          i--; break;
       }
       i++;
     }
     fclose (pFile);
   }
   return 0;
}
```



```
fund
email
exceed
quota
limit
account
-999

--------------------------------
Process exited after 0.01082 seconds with return value 0
請按任意鍵繼續 . . .
```

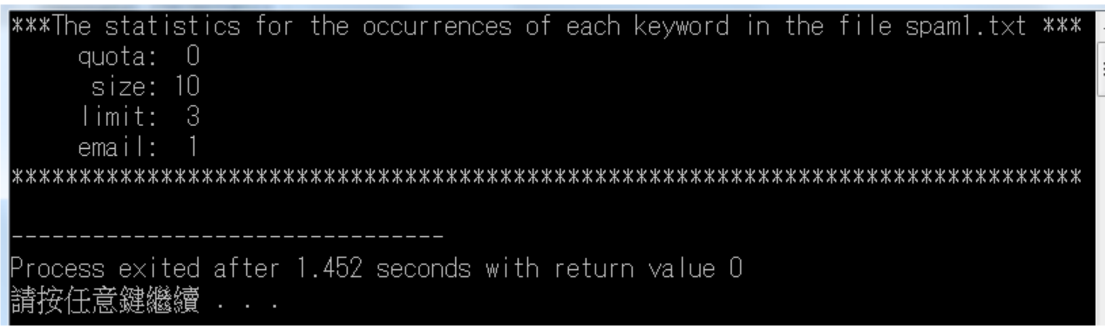
Main features:

(a) Your program has several features: (1) compute the number of occurrences for each keyword in an email and report the statistics; (2) report your judgement.

(b) For feature (1), the program needs to read in a keyword file and an email file. Run your program like: (due tonight)

chkspam –k keyword_file_name –e email_file_name –oc

where –k means the following parameter specifies a keyword file and –n means the following parameter specifies an email file.



```
***The statistics for the occurrences of each keyword in the file spam1.txt ***
     quota:  0
      size: 10
     limit:  3
     email:  1
*******************************************************************************

--------------------------------
Process exited after 1.452 seconds with return value 0
請按任意鍵繼續 . . .
```

(c) Enhance feature (1) as (due in 12/7):

chkspam –kn kFile_1 kFile_2 … kFile_n –em eFile_1 eFile_2 … eFile_m -oc

where –kn means the following *n* parameters are *n* keyword files, -em means the following *m* parameters are *m* email files and –oc means the execution of reporting occurrence statistics. This feature can read many keyword and email files sequentially, merge all keywords as a single set, and identify the occurrences of each keyword in all email files. If you fail in opening any file, you have to prompt the failure message but the program must continue to go.

(d) For feature (2), use another option to check if an email file is a spam email. (due in 12/7)

chkspam –kn kFile_1 kFile_2 … kFile_n -em eFile_1 … eFile_m –spam

You can read one or several keyword files to one or several emails. You have to determine a threshold value as spam lower bound, which implies that if the total cost for an email exceeds that threshold value then that email is regarded as a spam email. You may also need to adjust the weight for some keywords to improve the correctness of your judgement.

(e) Enhance your keyword library and derive another new way, such as the use of sub-sentence checking, to improve the feasibility and correctness of your program to identify spam emails. (due in 12/7)