Assignment 1

cpe 456 Fall 2012

Three may keep a secret, if two of them are dead.
— Benjamin Franklin
—`/usr/games/fortune`

Due by 11:59:59pm, Wednesday, Oct 24th This assignment is to be done individually.

Programs: `vig` and `ftable`

This assignment requires you to write two fairly quick programs. The first is to encipher text using the Vigenère cipher, and the second is a start at breaking ciphers, frequency analysis. Unfortunately, at this point, the cipher is stronger than your breaking tools, but we'll remedy that with the next assignment.

Enciphering: `vig`

The Vigenère cipher is a polyalphabetic cipher where different alphabets are used to encipher different letters of the plaintext in order to diguise letter frequencies. Each letter of the key determines the starting point for a shifted alphabet. Each letter of the plaintext P_i is enciphered using the letter of the key appropriate to it.

Enciphering Hamlet's soliloquy using the key "HAM" which generates the tableau in Figure 1:

Plaintext(P):	T	O	B	E	O	R	N	O	T	T	O	B	E
Key(K):	H	A	M	H	A	M	H	A	M	H	A	M	H
Ciphertext(C):	A	O	N	L	O	D	U	O	F	A	O	N	L

Plaintext:	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
H	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
A	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
M	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

Figure 1: A (sideways) Vigenère tableau for key "HAM"

`Vig` enciphers text using a Vigenère cipher. Details:

- As command line options, `vig` takes a string to use as a key, and optional input and output filenames. If either filename is not given, stdin or stdout is used (as appropriate). If the "`-d`" option is present, `vig` should decipher rather than encipher its input. You may implement a "`-v`" option for verbosity if you like.

  ```
  usage:  vig [ -v ] [ -d ] key [ infile [ outfile ] ]
  ```

- `vig` will encipher its entire input until it reaches the end of file[1]. It may not place any limit on the size of its input.
- `vig` must check its command-line options for validity. If they are invalid—e.g, a key that is not all letters—it should print a usage message like the one above and terminate with nonzero status.

[1] To indicate EOF while working from a terminal, type a Ctrl-D (ˆD) at the beginning of a line.

- If `vig` succeeds, its exit status should be zero.
- The cipher only operates on normal alphabetic characters ("A–Za–z"), and all lower-case letters are mapped to upper-case.
- Anything that is not "A–Za–z" in the input is passed through unchanged. (Were this a real enciphering tool, these characters would either be enciphered, too, or dropped.)

Cryptanalysis: `ftable`

`Ftable` generates a frequency tables for a given text or a subset of that text.

Details:

- As command line options, `ftable` takes:
  - an optional "`-v`" flag to increase verbosity;
  - an optional "`-s` *n*" option to cause `ftable` to skip the first *n* characters of the input (defaults to 0);
  - an optional "`-p` *n*" (period) option to cause `ftable` to only count every *n*th character (defaults to 1); and
  - optional input and output filenames.

  If filename(s) are not given, stdin or stdout are used (as appropriate).

  ```
  usage:  ftable [ -v ] [ -s num ] [ -p num ] [ infile [ outfile ] ]
  ```
- Only frequencies for alphabetic characters ("A–Za–z") are computed.
- All other characters are ignored. (i.e., they do not count for skip or period, either.)
- Lowercase letters are mapped to uppercase.
- Just like `vig`, `ftable` reads until EOF. There is no reason to make any assumptions about the size of the input except that your counts will not overflow an integer.
- When complete, the output will be:
  - The total number of characters tabulated, reported as "Total chars: *n*", where *n* is the number,
  - A line for each letter of the alphabet, containing the following space-separated fields:
    - The character, followed by a colon,
    - The number of those characters seen, right justified in a field of 9 characters,
    - The percentage of the total characters, aligned in parentheses to two digits of precision, and
    - A histogram containing one asterisk per percentage point or fraction thereof (the integer ceiling).

`int isalpha(3)`
`int tolower(3)`
`int toupper(3)`
	Functions defined in `ctype.h` that are useful when manipulating characters. Note, in non-U.S. locales, `isalpha()` might be true for characters other than “A–Za–z”. For this assignment, it is ok to trust that we are in the U.S.

int fseek(3)	Stdio version of `lseek(2)`. Useful for repositioning streams.
`double ceil(3)`	Math library function for computing ceilings. Remember to link with the math library (`-lm`)

Table 1: Some potentially useful library functions

Tricks and Tools

Some thoughts and advice that may or may not help you with this.

1. Some potentially useful functions are shown in Table 1. In addition, “man string” will show you many other useful string manipulation functions.

2. Note, because this cipher operates on a single character at a time, there is no reason ever to hold more than one letter of the plaintext or ciphertext in memory. (This is clearly also true of `ftable`)

3. For `vig`, be careful with your counts when you pass non-letter characters. These do not consume letters of the key. This would be a good thing to test while doing your incremental development.

Language

You can use C, C++, or Java, so long as:

1. The Makefile (below) builds the program appropriately, and

2. If you choose Java, you include scripts named `ftable` and `vig` that run your programs with the appropriate arguments when called by those names. (I don’t want to have to guess whether to run “`vig`” or “`java vig`”.)

Coding Standards and Make

See the pages on coding standards and make on the cpe 456 class web page.

Remember, this is a computer security class. Coding quality matters.

What to turn in

Submit via `handin` in the CSL to the `asgn1` directory of the `pn-cs456` account:

- your well-documented source files.
- A makefile (called `Makefile`) that will build your programs with “`make all`” or just “`make`”.
- A README file that contains:
  - Your name.

- Any special instructions for running your program.
- Any other thing you want me to know while I am grading it.

The README file should be **plain text,** i.e, **not a Word document**, and should be named “README”, all capitals with no extension.

Sample runs

Below are some sample runs of `vig` and `ftable`.

```
% vig d
I came I saw I conquered
L FDPH L VDZ L FRQTXHUHG
% vig ham
to be or not to be
AO NL OD UOF AO NL
% vig ham
TO BE OR NOT to BE
AO NL OD UOF AO NL
% cat > ham.in
to be or not to be
% vig ham ham.in
AO NL OD UOF AO NL
% vig -d ham
AO NL OD UOF AO NL
TO BE OR NOT TO BE
% vig ham ham.in | vig -d ham
TO BE OR NOT TO BE
```

```
% ftable ham.in
Total chars: 13
A:         0 (  0.00%)
B:         2 ( 15.38%) ****************
C:         0 (  0.00%)
D:         0 (  0.00%)
E:         2 ( 15.38%) ****************
F:         0 (  0.00%)
G:         0 (  0.00%)
H:         0 (  0.00%)
I:         0 (  0.00%)
J:         0 (  0.00%)
K:         0 (  0.00%)
L:         0 (  0.00%)
M:         0 (  0.00%)
N:         1 (  7.69%) ********
O:         4 ( 30.77%) *******************************
P:         0 (  0.00%)
Q:         0 (  0.00%)
R:         1 (  7.69%) ********
S:         0 (  0.00%)
T:         3 ( 23.08%) ************************
U:         0 (  0.00%)
V:         0 (  0.00%)
W:         0 (  0.00%)
X:         0 (  0.00%)
Y:         0 (  0.00%)
Z:         0 (  0.00%)
% vig ham ham.in | ftable
Total chars: 13
A:         2 ( 15.38%) ****************
B:         0 (  0.00%)
C:         0 (  0.00%)
D:         1 (  7.69%) ********
E:         0 (  0.00%)
F:         1 (  7.69%) ********
G:         0 (  0.00%)
H:         0 (  0.00%)
I:         0 (  0.00%)
J:         0 (  0.00%)
K:         0 (  0.00%)
L:         2 ( 15.38%) ****************
M:         0 (  0.00%)
N:         2 ( 15.38%) ****************
O:         4 ( 30.77%) *******************************
P:         0 (  0.00%)
Q:         0 (  0.00%)
R:         0 (  0.00%)
S:         0 (  0.00%)
T:         0 (  0.00%)
U:         1 (  7.69%) ********
V:         0 (  0.00%)
W:         0 (  0.00%)
X:         0 (  0.00%)
Y:         0 (  0.00%)
Z:         0 (  0.00%)
```

```
% vig ham ham.in ham.out
% ftable ham.out
Total chars: 13
A:         2 ( 15.38%) ****************
B:         0 (  0.00%)
C:         0 (  0.00%)
D:         1 (  7.69%) ********
E:         0 (  0.00%)
F:         1 (  7.69%) ********
G:         0 (  0.00%)
H:         0 (  0.00%)
I:         0 (  0.00%)
J:         0 (  0.00%)
K:         0 (  0.00%)
L:         2 ( 15.38%) ****************
M:         0 (  0.00%)
N:         2 ( 15.38%) ****************
O:         4 ( 30.77%) *******************************
P:         0 (  0.00%)
Q:         0 (  0.00%)
R:         0 (  0.00%)
S:         0 (  0.00%)
T:         0 (  0.00%)
U:         1 (  7.69%) ********
V:         0 (  0.00%)
W:         0 (  0.00%)
X:         0 (  0.00%)
Y:         0 (  0.00%)
Z:         0 (  0.00%)
% ftable -s 0 -p 3 ham.out
Total chars: 5
A:         2 ( 40.00%) ****************************************
B:         0 (  0.00%)
C:         0 (  0.00%)
D:         0 (  0.00%)
E:         0 (  0.00%)
F:         0 (  0.00%)
G:         0 (  0.00%)
H:         0 (  0.00%)
I:         0 (  0.00%)
J:         0 (  0.00%)
K:         0 (  0.00%)
L:         2 ( 40.00%) ****************************************
M:         0 (  0.00%)
N:         0 (  0.00%)
O:         0 (  0.00%)
P:         0 (  0.00%)
Q:         0 (  0.00%)
R:         0 (  0.00%)
S:         0 (  0.00%)
T:         0 (  0.00%)
U:         1 ( 20.00%) ********************
V:         0 (  0.00%)
W:         0 (  0.00%)
X:         0 (  0.00%)
Y:         0 (  0.00%)
Z:         0 (  0.00%)
```

```
% ftable -s 1 -p 3 ham.out
Total chars: 4
A:         0 (  0.00%)
B:         0 (  0.00%)
C:         0 (  0.00%)
D:         0 (  0.00%)
E:         0 (  0.00%)
F:         0 (  0.00%)
G:         0 (  0.00%)
H:         0 (  0.00%)
I:         0 (  0.00%)
J:         0 (  0.00%)
K:         0 (  0.00%)
L:         0 (  0.00%)
M:         0 (  0.00%)
N:         0 (  0.00%)
O:         4 (100.00%) ****************************************************************************************************
P:         0 (  0.00%)
Q:         0 (  0.00%)
R:         0 (  0.00%)
S:         0 (  0.00%)
T:         0 (  0.00%)
U:         0 (  0.00%)
V:         0 (  0.00%)
W:         0 (  0.00%)
X:         0 (  0.00%)
Y:         0 (  0.00%)
Z:         0 (  0.00%)
% ftable -s 2 -p 3 ham.out
Total chars: 4
A:         0 (  0.00%)
B:         0 (  0.00%)
C:         0 (  0.00%)
D:         1 ( 25.00%) *************************
E:         0 (  0.00%)
F:         1 ( 25.00%) *************************
G:         0 (  0.00%)
H:         0 (  0.00%)
I:         0 (  0.00%)
J:         0 (  0.00%)
K:         0 (  0.00%)
L:         0 (  0.00%)
M:         0 (  0.00%)
N:         2 ( 50.00%) **************************************************
O:         0 (  0.00%)
P:         0 (  0.00%)
Q:         0 (  0.00%)
R:         0 (  0.00%)
S:         0 (  0.00%)
T:         0 (  0.00%)
U:         0 (  0.00%)
V:         0 (  0.00%)
W:         0 (  0.00%)
X:         0 (  0.00%)
Y:         0 (  0.00%)
Z:         0 (  0.00%)
%
```