Assessed Exercise 2 Report

Student Name: Antonia Chalka

Student Number: 2138645C

Assumptions & Deficiencies

The user does not need to input the keyword in all capitals (the program capitalises the keyword in itself).

The program does not check if a file already exists with the output filename, so the user needs to be aware of the possibility of accidentally overwriting a file when encoding/decoding.

Summary: Program meets all specifications, including implementation of Vigenère cipher. Code snippets that were used for printing the cipher arrays for testing are at the bottom of their respective cipher class and commented out.

Testing

Filenames

Program will print appropriate message if file is to be encoded/decoded and will prompt user to re-enter input filename if it cannot determine the latter.

```
Please enter your filename:
undefined
Could not determine if file is to be encoded or decoded. Please retry.
Please enter your filename:
EncodeP
File EncodeP will be encoded. Output file: EncodeC
```

```
Please enter your filename:
CodeC
File CodeC will be decoded. Output file: CodeP
```

If the input file does not exist:

```
Please enter your filename:
madeupfileP
File madeupfileP will be encoded. Output file: madeupfileC
Please enter your keyword:
welp
Your keyword is: WELP
Monoalphabetic Cipher Array/Alphabet:WELPABCDFGHIJKMNOQRSTUVXYZ
Error when decoding/encoding file.
java.io.FileNotFoundException: madeupfileP.txt (The system cannot find the file specified)
        at java.io.FileInputStream.open0(Native Method)
        at java.io.FileInputStream.open(Unknown Source)
        at java.io.FileInputStream.<init>(Unknown Source)
        at java.io.FileInputStream.<init>(Unknown Source)
        at java.io.FileReader.<init>(Unknown Source)
        at Main.processFile(Main.java:109)
        at Main.main(Main.java:23)
```

Duplicate letters in keyword

Program capitalises, checks for repeated letters, and prints keyword. If the keyword contains repeated letter(s) it will ask the user to enter a new keyword.

```
Please enter your keyword:
TIGERRR
Your keyword (TIGERRR) contains a repeated letter. Please enter a different keyword:
TIGGER
Your keyword (TIGGER) contains a repeated letter. Please enter a different keyword:
tigger
Your keyword (TIGGER) contains a repeated letter. Please enter a different keyword:
test
Your keyword (TEST) contains a repeated letter. Please enter a different keyword:
asdfasdf
Your keyword (ASDFASDF) contains a repeated letter. Please enter a different keyword:
tiGger
Your keyword (TIGGER) contains a repeated letter. Please enter a different keyword:
tiger
Your keyword is: TIGER
```

Monoalphabetic Cipher

Encoding

Running the program (includes cipher array alphabet):

```
Please enter your filename:
CommonP
File CommonP will be encoded. Output file: CommonC
Please enter your keyword:
flamingo
Your keyword is: FLAMINGO
Monoalphabetic Cipher Array/Alphabet:FLAMINGOBCDEHJKPQRSTUVWXYZ
Finished encoding/decoding.
```

Input file (CommonP.txt):

```
THIS IS
A FILE
TO BE
ENCODED IN JAVA
WITH
SOME MORE
0123
TEXT AND
SOME
&&
SYMBOLS
**()))
THAT WILL
BE IGNORED!!
```

Output file (CommonC.txt):

```
TOBS BS
F NBEI
TK LI
IJAKMIM BJ CFVF
WBTO
SKHI HKRI
0123
TIXT FJM
SKHI
&&
SYHLKES
```

```
**()))
TOFT WBEE
LI BGJKRIM!!
```

Letter Frequency file (CommonF.txt):

```
LETTER ANALYSES OF CommonC FILE.

Letter    Freq    Freq%    AvgFreq%       diff
  A          1      1.3         8.2       -6.9
  B          7      9.3         1.5        7.8
  C          1      1.3         2.8       -1.5
  D          0      0.0         4.3       -4.3
  E          4      5.3        12.7       -7.4
  F          5      6.7         2.2        4.5
  G          1      1.3         2.0       -0.7
  H          4      5.3         6.1       -0.8
  I         10     13.3         7.0        6.3
  J          4      5.3         0.2        5.1
  K          7      9.3         0.8        8.5
  L          3      4.0         4.0        0.0
  M          4      5.3         2.4        2.9
  N          1      1.3         6.7       -5.4
  O          3      4.0         7.5       -3.5
  P          0      0.0         1.9       -1.9
  Q          0      0.0         0.1       -0.1
  R          2      2.7         6.0       -3.3
  S          6      8.0         6.3        1.7
  T          7      9.3         9.1        0.2
  U          0      0.0         2.8       -2.8
  V          1      1.3         1.0        0.3
  W          2      2.7         2.4        0.3
  X          1      1.3         0.2        1.1
  Y          1      1.3         2.0       -0.7
  Z          0      0.0         0.1       -0.1

The most frequent letter is I at 13.3%
```

Decoding

Running the program (includes cipher array alphabet):

```
Please enter your filename:
CommonC
File CommonC will be decoded. Output file: CommonD
Please enter your keyword:
FLAMINGO
Your keyword is: FLAMINGO
Monoalphabetic Cipher Array/Alphabet:FLAMINGOBCDEHJKPQRSTUVWXYZ
Finished encoding/decoding.
```

Input file (CommonC.txt):

```
TOBS BS
F NBEI
TK LI
IJAKMIM BJ CFVF
WBTO
SKHI HKRI
```

```
0123
TIXT FJM
SKHI
&&
SYHLKES
**()))
TOFT WBEE
LI BGJKRIM!!
```

Output file (CommonD.txt):

```
THIS IS
A FILE
TO BE
ENCODED IN JAVA
WITH
SOME MORE
0123
TEXT AND
SOME
&&
SYMBOLS
**()))
THAT WILL
BE IGNORED!!
```

Letter Frequency File (CommonF.txt):

```
LETTER ANALYSES OF CommonD FILE.

Letter   Freq   Freq%   AvgFreq%      diff
  A       5      6.7       8.2        -1.5
  B       3      4.0       1.5         2.5
  C       1      1.3       2.8        -1.5
  D       4      5.3       4.3         1.0
  E      10     13.3      12.7         0.6
  F       1      1.3       2.2        -0.9
  G       1      1.3       2.0        -0.7
  H       3      4.0       6.1        -2.1
  I       7      9.3       7.0         2.3
  J       1      1.3       0.2         1.1
  K       0      0.0       0.8        -0.8
  L       4      5.3       4.0         1.3
  M       4      5.3       2.4         2.9
  N       4      5.3       6.7        -1.4
  O       7      9.3       7.5         1.8
  P       0      0.0       1.9        -1.9
  Q       0      0.0       0.1        -0.1
  R       2      2.7       6.0        -3.3
  S       6      8.0       6.3         1.7
  T       7      9.3       9.1         0.2
  U       0      0.0       2.8        -2.8
  V       1      1.3       1.0         0.3
  W       2      2.7       2.4         0.3
  X       1      1.3       0.2         1.1
  Y       1      1.3       2.0        -0.7
  Z       0      0.0       0.1        -0.1
```

```
The most frequent letter is E at 13.3%
```

Vigenère Cipher

Encoding

Running the program (includes cipher array):

```
Please enter your filename:
VigenereTestP
File VigenereTestP will be encoded. Output file: VigenereTestC
Please enter your keyword:
tIger
Your keyword is: TIGER
Vigenere Cipher Alphabet:
TUVWXYZABCDEFGHIJKLMNOPQRS
IJKLMNOPQRSTUVWXYZABCDEFGH
GHIJKLMNOPQRSTUVWXYZABCDEF
EFGHIJKLMNOPQRSTUVWXYZABCD
RSTUVWXYZABCDEFGHIJKLMNOPQ
Finished encoding/decoding.
```

Input file (VigenereTestP.txt):

```
HELLO WORLD! HELLO EVERYONE!

non-capitals letter get ignored, for example:
hello my baby, hello my honey,
HELLO MY RAGTIME GAL!

SYMBOLS AND LETTERS GET IGNORED TOO:
1234567890
!"£$%^&*()_+=

GOODBYE CRUEL WORLD!
```

Output file (VigenereTestC.txt):

```
AMRPF PWXPU! AMRPF XDKVPHVK!

non-capitals letter get ignored, for example:
hello my baby, hello my honey,
LVETU QP KIMXZFM MEC!

LGSFFEA GRU EMZXVKA MIK BOTSIXL ZSF:
1234567890
!"£$%^&*()_+=

ZWUHSRM IVLXT CSIEL!
```

Letter Frequency File (`VigenereTestF`.txt):

```
LETTER ANALYSES OF VigenereTestC FILE.

Letter   Freq   Freq%   AvgFreq%      diff
  A       4      4.6       8.2        -3.6
  B       1      1.1       1.5        -0.4
  C       2      2.3       2.8        -0.5
  D       1      1.1       4.3        -3.2
  E       5      5.7      12.7        -7.0
  F       6      6.9       2.2         4.7
  G       2      2.3       2.0         0.3
```

```
  H       2       2.3       6.1        -3.8
  I       5       5.7       7.0        -1.3
  J       0       0.0       0.2        -0.2
  K       5       5.7       0.8         4.9
  L       5       5.7       4.0         1.7
  M       8       9.2       2.4         6.8
  N       0       0.0       6.7        -6.7
  O       1       1.1       7.5        -6.4
  P       6       6.9       1.9         5.0
  Q       1       1.1       0.1         1.0
  R       4       4.6       6.0        -1.4
  S       5       5.7       6.3        -0.6
  T       3       3.4       9.1        -5.7
  U       4       4.6       2.8         1.8
  V       5       5.7       1.0         4.7
  W       2       2.3       2.4        -0.1
  X       6       6.9       0.2         6.7
  Y       0       0.0       2.0        -2.0
  Z       4       4.6       0.1         4.5

The most frequent letter is M at 9.2%
```

Decoding

Running the program (includes cipher array):

```
Please enter your filename:
VigenereTestC
File VigenereTestC will be decoded. Output file: VigenereTestD
Please enter your keyword:
TIGER
Your keyword is: TIGER
Vigenere Cipher Alphabet:
TUVWXYZABCDEFGHIJKLMNOPQRS
IJKLMNOPQRSTUVWXYZABCDEFGH
GHIJKLMNOPQRSTUVWXYZABCDEF
EFGHIJKLMNOPQRSTUVWXYZABCD
RSTUVWXYZABCDEFGHIJKLMNOPQ
Finished encoding/decoding.
```

Input file (VigenereTestC.txt):

```
AMRPF PWXPU! AMRPF XDKVPHVK!

non-capitals letter get ignored, for example:
hello my baby, hello my honey,
LVETU QP KIMXZFM MEC!

LGSFFEA GRU EMZXVKA MIK BOTSIXL ZSF:
1234567890
!"£$%^&*()_+=

ZWUHSRM IVLXT CSIEL!
```

Output file (VigenereTestD.txt):

```
HELLO WORLD! HELLO EVERYONE!

non-capitals letter get ignored, for example:
```

```
hello my baby, hello my honey,
HELLO MY RAGTIME GAL!

SYMBOLS AND LETTERS GET IGNORED TOO:
1234567890
!"£$%^&*()_+=

GOODBYE CRUEL WORLD!
```

Letter Frequency file (`VigenereTestF`.txt):

```
LETTER ANALYSES OF VigenereTestD FILE.

Letter   Freq   Freq%   AvgFreq%      diff
  A       3      3.4       8.2        -4.8
  B       2      2.3       1.5         0.8
  C       1      1.1       2.8        -1.7
  D       5      5.7       4.3         1.4
  E      13     14.9      12.7         2.2
  F       0      0.0       2.2        -2.2
  G       5      5.7       2.0         3.7
  H       3      3.4       6.1        -2.7
  I       2      2.3       7.0        -4.7
  J       0      0.0       0.2        -0.2
  K       0      0.0       0.8        -0.8
  L      12     13.8       4.0         9.8
  M       3      3.4       2.4         1.0
  N       3      3.4       6.7        -3.3
  O      12     13.8       7.5         6.3
  P       0      0.0       1.9        -1.9
  Q       0      0.0       0.1        -0.1
  R       7      8.0       6.0         2.0
  S       3      3.4       6.3        -2.9
  T       5      5.7       9.1        -3.4
  U       1      1.1       2.8        -1.7
  V       1      1.1       1.0         0.1
  W       2      2.3       2.4        -0.1
  X       0      0.0       0.2        -0.2
  Y       4      4.6       2.0         2.6
  Z       0      0.0       0.1        -0.1

The most frequent letter is E at 14.9%
```