25th Annual Rowan University Programming Contest

hosted by the
Computer Science Department

Friday, 8 April 2011

Contest Problem

This problem combines two methods of encryption, the *transposition* and the *substitution* ciphers.

1 Transposition Ciphers

A *transposition cipher* rearranges the letters according to some pattern known by intended recipient. One simple method is to break a message into 25-character blocks, and write the 25 characters across in 5 rows. Then one reads down the columns in an order specified by a keyword, to get new 5-character pieces. The column order depends on the order that the letters in the keyword appear in the alphabet. Here is an example done with the first 25 letters of the alphabet and the keyword 'SNAPE' (notice the bold-faced letters):

Original (written across)					Transposed (read original down order by keyword)					
A	B	**C**	D	E	**C**	**H**	**M**	**R**	**W**	← keyword 'A'
F	G	**H**	I	J	E	J	O	T	Y	← keyword 'E'
K	L	**M**	N	O	B	G	L	Q	V	← keyword 'N'
P	Q	**R**	S	T	D	I	N	S	X	← keyword 'P'
U	V	**W**	X	Y	A	F	K	P	U	← keyword 'S'
Keyword:	S	N	A	P	E					
Column Order:	5	3	1	4	2	← alphabetical order of keyword letters				

The 1st row in the transposed version is the 3rd column of the original version; that column became the 1st row because 'A' (the 3rd letter in the keyword) is the alphabetically 1st letter in the keyword. The 2nd row in the transposed version is the 5th column in the original, because the 5th letter of the keyword, 'E', is 2nd alphabetically. The final result of this transposition is CHMRWEJOTYBGLQVDINSXAFKPU.

To decode, the text is broken into a grid again, and the rows are re-arranged into columns based on the keyword. The 1st row maps to the alphabetically first character in the keyword, and is moved into place based on where that letter appears. In this example, the row "CHMRW" becomes column 3, because the alphabetically-first letter in the keyword, 'A', is in the third position.

1.1 Notes

1. If a keyword is longer than five letters, only the first five should count. The keyword 'HOGWARTS' would be be used as if it were 'HOGWA'.
2. If a keyword has repeated letters in it, duplicate letters are skipped. The keyword 'RIBBON' would be used as if it were 'RIBON'.
3. If a keyword is longer than five letters and has repeated letters, the first five unique letters are used after duplicates are eliminated. The keyword 'ELEPHANT' would be used as if it were 'ELPHA', skipping the second 'E'.
4. A keyword with fewer than five unique letters should not be used. The keyword 'BOOTS' should be regarded as if no keyword had been given at all.
5. If there is no transposition keyword at all, or if the transposition keyword has fewer than five unique letters, the keyword should be replaced with 'ABCDE'.
6. All characters are transposed, including spaces and punctuation.

1.2 Longer transpositions

Since this can only accommodate 25 characters at a time, longer messages must be broken into 25-character blocks. If a line of input has fewer than 25 characters, it is filled in with letters starting at the beginning of the alphabet to make a complete block.

For example, the message 'THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.' is 44 characters long. It would be broken into two blocks, and the latter block padded with the six characters from A-F, filling it out to 25. Then the 25-character chunks would be written across in groups of five, and read down per a keyword. Using the keyword 'HOGWARTS', it would give this (spaces have been replaced with underscores for clarity):

Block One

Original					Transposed				
T	H	E	_	Q	Q	_	N	_	S
U	I	C	K	_	E	C	O	O	M
B	R	O	W	N	T	U	B	_	J
_	F	O	X	_	H	I	R	F	U
J	U	M	P	S	_	K	W	X	P

Keyword:	H	O	G	W	A
Column Order:	3	4	2	5	1

Block Two

Original					Transposed				
_	O	V	E	R	R	_	_	A	F
_	T	H	E	_	V	H	Z	G	D
L	A	Z	Y	_	_	_	L	D	B
D	O	G	.	A	O	T	A	O	C
B	C	D	E	F	E	E	Y	.	E

Finally, the transposed characters (on the right in the diagram above) are read across, giving this: 'Q_N_SECOOMTUB_JHIRFU_KWXPR__AFVHZGD__LDBOTAOCEEY.E'.

2 Substitution Ciphers

In a substitution cipher, letters are replaced with other symbols (most commonly other letters, but any characters can be used). In order for this to be sensible to the person decoding the message, some systematic way to choose substitute letters is needed. One method is the *Cæsar cipher*.

In a Cæsar cipher, each letter is shifted down the alphabet, so using one with a shift of 3 in English would mean A is shifted to D, B is shifted to E, and so on. Letters at the end of the alphabet wrap around, giving this:

Original	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
Encoding	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

The message 'THIS IS A TEST.' would be encoded as 'WKLV LV D WHVW.', and the person who received the message would un-do the shift in order to understand it (for each D they would substitute A, and for each W they would substitute T, and so on). Non-alphabetic characters, such as spaces and digits and punctuation, would be left alone.

Substitution ciphers can be defeated by letter-frequency attacks; the most common letter in English is 'e', so one counts all the letters in the message and whichever one appears most often is quite likely 'e'. Knowing the most common letters, and looking for patterns (a word that appears

to be '`T_E`' is quite likely 'THE', giving another letter), makes it easier to decipher such messages. It is so easy that it cryptograms are a standard puzzle on newspaper comic pages.

2.1 Stronger Substitutions

The substitution specification for this problem involves using letter shifting, as in a Cæsar cipher, but instead of shifting each letter the same amount, the shifts depend on the letters of a keyword.

To determine the shifting, the alphabet is laid out in order and each letter is assigned a number based on its position:

Letters	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
Value	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

So $A = 1$, $B = 2$, $C = 3$, and so on up through $Z = 26$. Lower-case letters have the negative value of the corresponding uppercase letter. (So a = -1, and b = -2, ... z = -26.) The values for each letter of the keyword determine how far you shift each letter of the message. The keyword *Jayne* would have the values {10, -1, -25, -14, -5}.

Encoding a message is done by shifting each letter the number of spaces specified by the value of the letters in the keyword. Using the keyword *Jayne*, you would shift the first letter right 10 spaces, and the second letter left 1 space, and so on through the fifth letter, which would be shifted left 5 spaces. Starting at the first letter past the end of the keyword, you go back to the beginning; for this example, the sixth letter would be shifted right 10 spaces, the seventh would be shifted left 1 space, and so on until the entire message was enciphered.

Using the keyword *Jayne*, the message 'THIS IS A TEST.' would be enciphered as follows (spaces have been replaced with underscores for clarity):

Message	T	H	I	S	_	I	S	_	A	_	T	E	S	T	.
Keyword	J	a	y	n		e	J		a		y	n	e	J	
Shift	10	-1	-25	-14	–	-5	10	–	-1	–	-25	-14	-5	10	–
Result	D	G	J	E	_	D	C	_	Z	_	U	Q	N	D	.

Giving the final result 'DGJE DC Z UQND.'.

To decode, the enciphered letters are shifted in the opposite direction (so D is shifted left 10 characters, giving T, and G is shifted right 1 character, giving H).

2.2 Notes

1. Unlike transposition ciphers, lines are *not* padded out to any particular length. Keyword position must be maintained across line breaks. If there were a line after 'THIS IS A TEST.', the first letter would be encrypted with '**a**' (since the last letter of the previous line was encrypted with 'J'); the new line does *not* start over with '**J**'.

2. Unlike transposition keywords, substitution keywords have no minimum length and no difficulty with repeated letters. Also, non-alphabetic characters are passed unchanged, and do not count for determining position in the keyword.

3. With a one-character keyword, this is the same as a Cæsar cipher, because each letter is shifted the same amount. If the keyword was 'C', this would give the result in the diagram at the beginning of section 2.

3 The Combined Cipher For This Contest

Your challenge for this contest is to implement both transposition and substitution ciphers.

When encoding, your program will first transpose the characters as described in Section 1, and then substitute the characters as described in Section 2.1.

When decoding, your program will reverse the steps, undoing the substitution and then undoing the transposition.

4 Input

4.1 Input Specification

For text input, your program should accept input in the following format:

1. An integer, $\mathcal{D}$, where $1 \leq \mathcal{D} \leq 50$, which is the number of datasets in this file.
2. $\mathcal{D}$ data sets, each of which is in this format:
   (a) One line with a single word, which is the transposition keyword.
   If this keyword is 'PASS', do not do the transposition step.
   (b) One line with a single word, which is the substitution keyword.
   If this keyword is 'PASS', do not do the substitution step.
   (c) One line with a single integer, $\mathcal{E}$, which is the number of lines to be encoded.
   (d) $\mathcal{E}$ lines, each with plain English text to be converted to ciphertext.
   (e) One line with a single integer, $\mathcal{D}$, which is the number of lines to be decoded.
   (f) $\mathcal{D}$ lines, each with ciphertext, to be converted to plain English.

4.2 Sample Input #1

Data in file	Item # above:	Meaning in plain English
2	1	*this file has 2 data sets*
SNAPE	2a	*The transposition keyword for set 1 is* SNAPE
PASS	2b	*Data set 1 has no substitution step.*
1	2c	*There is only 1 line to be enciphered.*
ABCDEFGHIJKLMNOPQRSTUVWXY	2d	*The line to be converted.*
3	2e	*There are 3 lines to be deciphered.*
G RUTEE BIELOO WBEADHLFY ERAFKTTCHMRI.EJ SBGLAHYDI NPR CSO) I E (ERB HEV	2f	*The three lines to decipher.*
HOGWARTS	2a	*The transposition keyword for set 2 is* HOGWARTS
PASS	2b	*Data set 2 has no substitution step.*
2	2c	*There are two lines to be enciphered.*
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.	2d	*The lines to be converted.*
2	2e	*There are two lines to be deciphered.*
I LAF T EDNGNABOHOMCF:B.E AU T -O - LZ	2f	*The lines to decipher.*

(This input, corresponding to the examples in Section 1, will be on the website as **sample1.txt**.)

4.3 Sample Input #2

Data in file	Item # above:	Meaning in plain English
2	1	*this file has 2 data sets*
PASS	2a	*Data set 1 has no transposition step.*
Jayne	2b	*The substitution keyword for set 1 is* Jayne.
2	2c	*There are 2 lines to be enciphered.*
THIS IS A TEST. THIS IS LINE 2.	2d	*The lines to be encrypted.*
2	2e	*There are 2 lines to be deciphered.*
DGJE DC Z UQND. SIUN SR MUIO 2.	2f	*The lines to decipher.* *(Padded with spaces to fill out 25 chars.)*
PASS	2a	*Data set 2 has no transposition step.*
C	2b	*The substitution keyword for set 2 is* C.
2	2c	*There are two lines to be enciphered.*
ABCDEFGHIJKLM NOPQRSTUVWXYZ	2d	*The lines to be converted.*
0	2e	*There are no lines to be deciphered.*

(This input, corresponding to the examples in Section 2, will be on the website as **sample2.txt**.)

4.4 Sample Input #3

Data in file	Item # above:	Meaning in plain English
1	1	*This file has 1 data set.*
ZEBRA	2a	*The transposition keyword for set 1 is* ZEBRA.
banana	2b	*The substitution keyword for set 1 is* banana.
3	2c	*There are 3 lines to be enciphered.*
DILIGENCE IS THE MOTHER OF GOOD LUCK. -- BENJAMIN FRANKLIN	2d	*The lines to be encrypted.*
3	2e	*There are 3 lines to be deciphered.*
C BNQMDQSTLQ DCXBDMS IZD VQCBH MDNFD CMEFQYBTHA .D CHMM MEJ -I W AYQG -MZM	2f	*The lines to decipher.*

(This input will be on the website as **sample3.txt**.)

NOTES:

No line of input (and thus no keyword) will be more than 25 characters long.

No keyword will contain any non-alphabetic characters.

All lines to be deciphered will be exactly 25 characters long.

All text input to be enciphered or deciphered will be in ALL-UPPERCASE LETTERS.

You need not do error-checking on the input. Each line will have exactly the number of items described with no stray characters. There will be no blank lines.

5 Output

5.1 Output Specification

For each data set configuration, your program must generate output as follows:

1. The text '`Analyzing D data sets`', where **D** is the number of data sets in the input.
2. The text '`Data set S:`', where **S** is the number of the data set being reported on.
3. The text '`Transpose Keyword TK:`', where **TK** is the transposition keyword for this data set.
4. The text '`Substitute Keyword SK:`', where **SK** is the substitution keyword for this data set.
5. The text '`Enciphering E line(s):`', where **E** is the number of lines to be enciphered.
6. **E** lines of output, each the encrypted version of the original plain text input lines.
7. The text '`Deciphering D line(s):`', where **D** is the number of lines to be deciphered.
8. **D** lines of output, each one the decrypted version of the enciphered input lines.

5.2 Sample Output 1

```
Analyzing 2 Data Sets
 Data Set 1:
 Transpose Keyword: SNAPE
 Substitute Keyword: PASS
 Enciphering 1 line(s):
CHMRWEJOTYBGLQVDINSXAFKPU
 Deciphering 3 line(s):
DIG THE WELL BEFORE YOUAB
ARE THIRSTY.ABCDEFGHIJKLM
        (CHINESE PROVERB)
 Data Set 2:
 Transpose Keyword: HOGWARTS
 Substitute Keyword: PASS
 Enciphering 2 line(s):
Q N SECOOMTUB JHIRFU KWXP
R  AFVHZGD  LDBOTAOCEEY.E
 Deciphering 2 line(s):
NO FIGHT: NO BLAME.ABCDEF
               -- LAO TZU
```

(This output corresponds to Sample Input #1 from §4.2.)

5.3 Sample Output 2

```
Analyzing 2 Data Sets
 Data Set 1:
 Transpose Keyword: PASS
 Substitute Keyword: Jayne
 Enciphering 2 line(s):
DGJE DC Z UQND.
SIUN SR MUIO 2.
 Deciphering 2 line(s):
THIS IS A TEST.
THIS IS LINE 2.
 Data Set 2:
 Transpose Keyword: PASS
 Substitute Keyword: C
 Enciphering 2 line(s):
DEFGHIJKLMNOP
QRSTUVWXYZABC
 Deciphering 0 line(s):
```

(This output corresponds to Sample Input #2 from §4.3.)

5.4 Sample Output 3

```
Analyzing 1 Data Sets
 Data Set 1:
 Transpose Keyword: ZEBRA
 Substitute Keyword: banana
 Enciphering 3 line(s):
E GFAXB KQUME DGDFNMCCHQG
ATZFX  .DVEBJPHSKYEWNABAG
 QHMM  YEX -I W AKQU -MZM
 Deciphering 3 line(s):
THREE MAY KEEP A SECRETAB
IF TWO ARE DEAD.ABCDEFGHI
     -- BENJAMIN FRANKLIN
```

(This output corresponds to Sample Input #3 from §4.4.)

Your output does **not** have to duplicate the sample output as regards spacing or use of upper/lower case. Your output should be neat, but need not exactly match the sample.

6 Test Data

Run your program on this input and print the results. **You must submit printed output to earn full points.** Your program will also be run on data known only to the judges.

6.1 Test Input #1

(Substitution Test)

```
2
PASS
Zoe
5
ABCDEFGHIJKLMNOPQRSTUVWXY
"TO ERR IS HUMAN, BUT TO
REALLY FOUL THINGS UP YOU
NEED A COMPUTER."
           - PAUL EHRLICH
5
AMXDPAGSDJVGMYJPBMSEPVHSY
"EJ ECM ID CUXVN, MPT EJ
RPVLWT FZPL ECIYBS FK YZP
NPZD L XOXKUEZR."
           - AVUW ZHCGINC
PASS
Electroferromagnetic
4
THE BEST TIME TO PLANT A
TREE IS 20 YEARS AGO. THE
SECOND BEST TIME IS NOW.
        - CHINESE PROVERB
5
NT D EGL DCS PWFER MA
XNFM ICRK G BCYZ, Q'L
DCDGP ONV CNFNQ LWFL
CWCCF RAMMVVKNBB QNM LRZ.
        - IJCNGTY GOEZTZI
```

6.2 Test Input #2

(Transposition Test)

```
3
FIREFLY
PASS
2
THE FIVE BOXING WIZARDS
JUMP QUICKLY.
2
FBQLOWZEI OOSCS IUYTLM KP
 EABGTI EDHNG.EJDVCHEXRAF
SERENITY
PASS
2
TIME FOR SOME THRILLING
HEROICS.
2
ET HAOG CYG O P ATEPWTOTR
OHI . E RCNRMEBWE'HATW AO
ROWAN
PASS
2
TWENTY-FIFTH ANNUAL
PROGRAMMING CONTEST
2
AIICHNVTDIOURAFR EYEWNSBG
P NECUSCPDOEI ACTCETMREDB
```

6.3 Test Input #3

(Combination Test)

```
3
TTTTTEEEEEPPPPPIIIIIDDDDD
QWERTYuiop
3
A SHIP IS SAFE IN HARBOR,
BUT THAT IS NOT WHAT
SHIPS ARE FOR.
4
KHWMYNQN M'YVLIFIGDTO  OR
, BFXUAMPNVKOZWFTJFOP N Y
UTAKVCONZKUSZJUZ.PTFPJUBL
  HIM  PES -NHL -Q J  RLY
TARDIS
Gallifrey
3
THERE'S NOTHING MORE USE-
LESS THAN A LOCK WITH A
VOICE PRINT. -- BORUSA
3
VSMPR'LCXDAMFYUHCJBIC   ,
CZR V'WVOKSYMUYVJXHBR  ,O
FV.ME'JIFZHEWPHOSZTYU ELD
ROWAN
University
3
ERUDITIO SPES MUNDI
TRIVIUM AND QUADRIVIUM:
    LIBERATING THE MIND
2
CKVIMWVEJBZQISJEWLP    ZP
TLSOL HE!K5O ZN2RPOIAEITX
```

All sample and test data sets are available at <http://elvis.rowan.edu/rupc/2011>

You may choose to have your program read the input from the keyboard, or ask the user for a filename and then read the file. Users of GUI-based programming environments may prefer to use text boxes into which the values can be entered, and buttons to begin their calculation. Any reasonable variation in the spirit of the problem is acceptable.