Information Security Exercises

Chapter 1: Set one: Introduction. Deadline: Monday September 23, 9:00 (9:00 AM)

Exercise 1.
Purpose of this exercise: familiarization with an acceptable use policy.

Download (and submit a link to) the University of Groningen's Acceptable Use Policy (AUP) (submit the Dutch version if your primary language is Dutch, otherwise submit the English version)

- What (if any) are the differences between the responsibilities of 'ordinary' users and systems managers? Do systems managers have special privileges and responsibilities (if so, what are they)?
- What is the ground-rule upon which the RUG's AUP is based;
- Mention four advices for users of 'RuGnet';
- Describe four actions that are prohibited by the RUG's AUP;
- What sanctions can be applied to those who violate the AUP?
- If a sanction is applied to you, where can you go to challenge that sanction?

Exercise 2.
Purpose of this exercise: construct a tool to work with a substitution cipher

Write a program (**See the 'general instructions' document for more information about the program requirements**) showing the following usage information when started without any further arguments and that behaves accordingly when the appropriate arguments are provided:

```
Usage: substitution [-o] [-d] mapping
Where:
   -o: keep non-letters as is, honor letter casing
   -d: decrypt
   mapping: 26 letter char-mapping
            or an int-value
```

```
En/Decrypts stdin to stdout. Only letters are encrypted,
all other characters are silently ignored, unless -o was
specified, in which case they are used as-is.
When -o is specified, letter casing is honored, otherwise all
letters are converted to lower-case letters.
Use an int-value to do a letter shift (% 26, 0: a = a)
Shift 3 is the classical Caesar encryption
```

The text in the file `2019.enc` was encrypted using the -o option and the following character mapping:

```
    yxzwvtsqpnrmlkjgfdchbuoaie
```

Submit the decrypted text, and the source of your program.)

Exercise 3.
Purpose of this exercise: learn to break a simple substitution cipher text

Decrypt the following text, showing only the encrypted letters. Submit the decrypted text, as wel as the human readable decrypted text, i.e., a text in which you've put blanks between the individual words:

```
dlsjvtlavaoljvbyzlhivbapumvythapvuzljbypafaopzjvby
zlpzhivbazljbypunpumvythapvupuaopzjvualeadlaopurmv
ylehtwslhivbaovdavwylcluaaolbuhbaovypglkylhkpunvmp
umvythapvuvyhivbaovdavwylcluaaolbuhbaovypglktvkpmp
jhapvuvmpumvythapvulujyfwapvupzaolihzpjavvsmvyzljb
ypunpumvythapvuthuflujyfwapvutlaovkzlepzazvtlhsylh
kfaovbzhukzflhyzvskpupaphssfdlssmvjbzvuzptwsltlaov
kzavlujyfwapumvythapvumvssvdpunaopzdlssbzljohyhjal
ypzapjchsblzpkluapmfpunpumvythapvuthrpunpakpmmpjbs
aavtvkpmfpumvythapvubuuvapmplkshalypuaopzjvbyzldls
spuayvkbjlwlyzvuhslujyfwapvudhfzavjvuayvshjjlzzavp
umvythapvuhukovdavrllwfvbywypchjfdlovwlfvbdpssluqv
faopzjvbyzlhivbapumvythapvuzljbypaf
```

What shift was applied to the plain text to obtain the above cipher text?

If a shift `x` (as in `c = (p + x) % 26`) is applied to a plain text character `p` to obtain the cipher text character `c` then the original text can be retrieved from the ciphertext by undoing the shift by `x`. (e.g., by computing `p = (c - x) % 26`.

However, this may result in negative values for `p`. To prevent this, it is also possible to *encrypt* the above text by applying a shift `y` such that following the encryption of the above cipher text the resulting text is equal to the original text (so, here you're encrypting the encrypted text).

What is the smallest postive substitution cipher shift value of `y` returning the original text from the above cipher text?

Exercise 4.
Purpose of this exercise: learn to break a standard vigenere cipher text

The following text (encrypted from a passage of a book by Bruce Schneier) was encrypted using a standard vigenere cipher:

```
qptqkkwlckhkmzpnkmupkcvkwzpicacgelmfwhzlzfnmfmcksh
cmfgvnfmrzbtpgybokwilfqbgqujbppqgbrmjiumcgliiilzbrg
phqkwzhkbubwvvzlnmpmmbvigbqgvetsmlsuzxhagxkcbgrigj
qewarvyxzbrrpexezjikcbbuigmwbbmqnwoeoipmixrzzdxawh
gxkiuxyahkiykpbqqfascvvvpgcmmueehpsfmlkkpkcigbwkrd
xqlehtvzvzyvqxriigirqbgtxfdbbmfmlkgzhnpleeikqvrpvl
gultwytyuikoxeyqaxhczbaztbvojzizpizlerzkxgaphqslvb
aigbrmnqmfjbufukptpmvwitkqygmquchffxqiawxnvzxganem
tvjxrerxrzymfqqtgmlpqgebuxgudunlqptxofvxtmvlxnvmtt
mfwvugxxpauxerjwbqiohbgelayanwszkmwjqaxezkivfmqmsz
ymvmuznrotimgwalpoemlfmvlehcmmmqamixtmirbuxgudunlq
ptxofvmfmbgpenirrwckibvvmcdryvudtxyzabrmnptriybgkr
vwzwotvkkieiqazehfcmgagavuloayxexzkebbtmphytkmkkmn
lyxvmgazlixofvmfmexwtflxrmpmmuebaczrlruimlnwalizym
kcaahvojsfyvnzisvvmwwhaebvbhydbbhzymmfzrtxlfzwcknw
iynmaydrnwkubaganitxfivfbbvssgcmczfxgaiqmwerwvgnjh
vmftvulvwrprwmlwmkcvgipgpmkqiawpoemlzmgpikebacujxh
kwqgclvyjkimgrigmeibmkqmnoiyuzhnxrkwodxxpabgezfzlr
pvxzkjigbbuxmxtiiyjvemzzmlumhlivimocvgbzktwnlbrkqk
ranpmfemqvmgazlixofvtllnvgkjavmvgksrkwttwvwhownxpm
amxnimtravyaktigydbbhzymmfzrtxynmocebgmlnmvyvgpibv
thqbvfemzvxkgfnvviqlceuxrocmtpvrwxnrbmfmjhvrulhcaa
mauismfqfpeejwfcpvlxuigypwzmlkmqzcvrkiczsbnitxxnvn
bpagpirclhaczxrzvlwcapkmvkqhlwsttucgtjxutfkkqvaqca
ixnildwefyrrbxbjleiuejtrbvlxgrtuczgbexfcgbiawyyvlt
kmgtpizxaczqbwikwluqgvlhvbpcmavmvymkytcaehvblytoxv
zzalwagxquetrqevmgnvltjxutfkkatdbrkwkmmkytjhvjjigb
ajbxiymlumexmtuqvybrwfenzbrqazxnvtxrbrkslkpxawekiy
gwgbqazergptzmgbrzymvgxuxvzvfmjigxvoerhfiagiykzbrp
rfmajqgfqfpsxbxhjqtkevyqtgvixrzvlmfmgtfacikckgteii
qmgknegudxhlmamslkpxtqtxrxvkbnprkxnvbkgbuxqolavgxu
```

xvnfextmehrrpxkmdvwijrxkmoexwyzdxpqtbhgelipmqbgzrj
ecallxkdnhpajbxiyqgejrmakvvvgxuxvgcxayjrmwcyimgaah
aqewplifmlkmqzcvrkiizxaczjtwuiqzgvnepeumlazvuijsgz
gwitrhrbmgagtfkcttqwvglojjhmsytgowztbmylmmxqhtiaue
zkqlrioxprrahfmonmrkcimvgaizrjnjiexgzrwyrzvmlkdqnq
jhmejumwyzriigkqgekbnrzvzlgoatokpbhqevmgntqifmetpv
yiucbfxzkigecbgxvcymkciftphvzmgiawxxzbacuvnwajmwyn
vqijgimrmegsljcuqbvmyzzwgqjrepgjwlqkuxqkdmtlbgaivr
bmczahjyljlrqgnxofvlawhehhvmtqqyrgnrvzclfbqvcguwar
eiikqgeiaxaqvgdcgfpixvbrnqptprpabloyxauillmzfasxkx
apifxwqewplbbuszyxtpbvxwoeiwtiaviuibkyvffmzkmwmcgh
jhrvwytbgkczbarprfiyjizcjrepgjwlkmgasjkpnqzrjyoimw
qbehrmjmvszvmclfzhltlmlkbmryavmmyimeybvoirpmtqgghw
ktckcifasxksxwxukeyvatwjlttxvdbmcfivomimckbgzkiatr
qbgfkcttqwflcykmfuifvstjqwcznupedwkcarvyxvjeyqfxhk
mqzcvrkivljegauxhnzawcapkmvkqhlwstwodqeyzonxykzhlo
rkeakwdcgpbtnvzucnbkizymvmcemslymgpgvbmuwnkyvpxmtc
imczvgxnvbaamamyxpbacqaoitkqhlwsuircilmapbtnvzpyaz
bwgkbkgjhmijkwogorgixvlttqqdeneqgfqfusubbackbwihim
timelpgdmgrmqmlkdqlybgkmhlbbmvorwgpqgebutxnzammzla
ejzogmzrwxnzabkxbkxgebvmvgkmhlbbmvnghoeamciqgesvlt
pmtkiyjqociawirvuxlbnkcizxaczshvnzuogorgixvbamctal
kyiwlwgamtxbhbwjbxnzbmfmibkkemkckvilkiotgvrwexvxnr
igbstwwkzmvgkkokxnbvhrgctrqbehrmewmclnnxnfztllztxn
vutrqpbettptptrlpakebborwsjxahltrpmytikpwyeggctxbb
uxzoxmgczrvmvymksvokigbiujmvglojxbckrmlkrtifioxxiz
xaczvgeiyqebzrgwsrotxqaxmtjkbcvgbjotifczvvetumlazv
uijkpxtqtxrxvkbnprkeyzuimafbfrvwyrzngwrrbbmvgamyim
isbnmmueetqvbmhkjmktmqvlgitxqjnufgxmbqsahatkwaydru
vubmgydnkmgebhdbuxgogpxpifxexcgtqpbpibvzaclvwrzgcu
jqfalojehpsxtwojsbcvgbvkcgupwxxxnvkbnprketuxnztvll
kubacbrvltzyncqamlkkpvcvgnvevdxljrysxvbagagasaxplm
urlooctxbkertzrvtjgfmwifcebwpveyzwgytyrfxvidrprvmv
ymkgvgaizykxlbhkciigirwtkevyqvqtvwixltxsarweyrktjk
heezzwgyqquczymluqflexdgucbjxitrvwrprommvvxpmpbtnv
zbqavftrvmgmctaxusmtdqrehizxaczvymzzanqmqbrifvcsvp
mmueebrppbtnvzwgaxlxnvkhlnrwixrbxqbnmiyfntkmebggww
kcfnftrvclclnuvgjavgxuxvjzadrwvftrvuxlbgaibzoxlmex
gogpxplhkmtxbacizxvotigaqibpcrzmfmphrlvlxpiprwsval
yorlakimyyzskssjmvpmgtrjkpxsvvhrxvonjieeciiivimqml
kzzfcaftkkjbapwhzlulbmfmjtvzymvmvsxhkiimctrthkiaag
xckmsrzbjgexpovlnnwamlxvmdcgcavgjmlkiavlkjbxpjynjl
twfntrmibzkmmzltrjramfmjtviruxrwnvpujmvmurkiziqusb
vhrmztuczgoixeifrzvxhzfzxnivkxnvjkmsrggogpxpkexezz
vzrproixeiftqtxrkimvgxuxvoejnrvbfezkmkupnmlkuqwrpr
vmvymkuiflxoctostaxvgstxrwpkcvkigytllmymmklizlauis

```
ameroixvdxlbhtprptxbbbmlkfvxrqzxtguiipwitfrpcgzzrt
ogstxaqcaix
```

Only the letters of the alphabet were used, all uppercase letters were first converted to lower case. When computing the most likely key size use the following hints:

- The key length is between 5 and 15 characters long
- The used key is a single existing English word
- For a potential key size `k, k` vectors of letter frequencies are computed. Having computed these vectors, compute the standard deviations of the frequencies found in each of the `k` vectors and sum the `k` computed standard deviations. The correct key size will have the largest 'spike' (compared to its neighbors of potential keys of `k - 1` and `k + 1` characters).

  E.g., a particular encrypted text might have been encrypted with a key of size 8. The table of sums of standard deviations could look like this:

  ```
  Sum of  5 std. devs: 17.2,
  Sum of  6 std. devs: 24.9,
  Sum of  7 std. devs: 17.9,
  Sum of  8 std. devs: 40.9,
  Sum of  9 std. devs: 20,
  Sum of 10 std. devs: 27.5,
  Sum of 11 std. devs: 23.1,
  Sum of 12 std. devs: 37.7,
  Sum of 13 std. devs: 24.8,
  Sum of 14 std. devs: 29,
  Sum of 15 std. devs: 26.3,
  ```

  clearly showing that the spike at key length 8 is the highest.
- Estimate the most likely shift for each of the characters from the maximum frequency, which probably represents the character `e`. E.g., if the maximum is at character `k` then the associated keyword character is probably `g`. This is not a mathematical certainty, though. Sometimes there is a *tie* (e.g., the frequency at `k` equals 15, but the frequency at `s` as well, making `o` a good alternative character in the key), or the count may be off by one (e.g., `f('s') == 14`, making `g` the most likely key character, but `o` is a good candidate as well).
- Remember: no blanks, no punctuation, no numbers appear in the encrypted text.
- Compute the standard deviation of a vector of frequencies as follows

```
sqrt( sum(x * x) / 26 - sqr( sum(x) / 26 ) )
```

with `x` a letter frequency, `sum` a function iterating over all `x` values, `sqrt` the square root function, `sqr` the function returning the square of its argument (i.e., `sqr(y) = y * y`). Since there are 26 letters in the alphabet, the fixed denominator 26 was used in the above formula.

Submit:

- Your decrypted text; put appropriate blanks between the words
- Your table of sums of standard deviations for each of the probed key sizes.
- The initial keyword suggested by the vigenere-breaking algorithm
- All reasonable alternatives. Let's consider 'reasonable alternatives' the next two best candidates for each of the characters of the key, given the length of the key you eventually use. For a given key length each subsequent character of the key *probably* is the most likely key character. Sometimes, however, that doesn't come up with an existing word. In those cases the next most likely key character could be considered. Thus it's usually possible to determine which key characters meet the requirements (e.g., an English word). So if the key length is 4 then you might find:

key character:	4	3	2	1
most likely:	d	x	i	d
2nd best:	p	e	q	b
3rd	z	m	a	f

  In which case the best bet 'dxid' clearly isn't a word, but using e for x and a for i produces 'dead' which is an English word. For the exercise also submit a table like the one shown above (but then of course for the key length you actually use)
- If you had to generalize the vigenere cipher in such a way that not only the (lowercase) letters were used in the encryption process but all printable ascii characters, what would be your most frequently occurring character in that case?