1. (Do not submit) Download a cryptography library from the Internet and install it on your computer. You may use any library you find, for your favorite programming language. For this exercise you only need DES support. However, for future exercises, it is recommended that you find a library (or libraries) that support the following cryptographic algorithms:

   DES, Triple-DES, AES (Rijndael), RSA, Diffie-Hellman, El-Gamal, HMAC-MD5

   • A good starting point for searching is

   • A library for the C (or C++) language is Cryptlib, from http://www.cs.auckland.ac.nz/~pgut001/cryptlib/, with source code and a 350 page manual. It’s a bit complicated to use.

   • For C++ (not C) you can use Crypto++, from http://www.cryptopp.com/.

   • Another possibility is the GNU libgcrypt, available from http://directory.fsf.org/project/libgcrypt/. libgcrypt is built for Linux - compiling on Windows may be tricky (google may find recipes).

2. • Use the crypto library you installed in question 1 to encrypt your Teudat Zehut number (expressed by 8 ascii digits, without the rightmost checksum digit) using DES, with the key $K_0=\text{aa aa aa aa aa aa aa aa}$ (in hex). Call the result $C_0$.

   • For bit positions $i=1,\ldots,64$, flip bit position $i$ of $K_0$ to obtain $K_i$ (go through 2a-aa-..., then ea-aa-..., etc.). Encrypt your teudat zehut number again using the key $K_i$ to get $C_i$. Compute the Hamming distance between $C_i$ and $C_0$ (i.e., compute the number of bit positions that are different between $C_i$ and $C_0$).

   • The answer to the question is two values: (a) the minimal (non-zero) Hamming distance among all positions $i$: the minimal number of cipher bits that change due to flipping a single key bit. (b) the maximal Hamming distance among all positions $i$.

Example, using teudat zehut number of 12345678:

```
# ./des-weight -e 12345678
Basic key: aa aa aa aa aa aa aa
Ciphertext: f3 c3 f7 a3 b5 9d 88 4b
...  
Key: a8 aa aa aa aa aa aa
Ciphertext: 37 43 1e 43 51 b4 ec
Bit difference: 33
...  
```
Notes:

• 8 of the 64 bit positions in the key are ignored by DES so the ciphertext \( C_i \) will be identical to \( C_0 \) and the Hamming distance will be 0. Skip these bit positions.

• It is simplest to set the DES mode to be ECB. (in some libraries the default setting is CBC mode - that’s OK too but read the next item).

• In CBC mode you need to supply an IV (initialization vector). If you work in this mode then use hex 00 00 00 00 00 00 00 00 (8 binary zero bytes). In ECB mode there is no IV.

• Make sure you use hex aa-aa-aa-aa-aa-aa-aa-aa as a raw DES key; do not let the crypto library process it. Use the example above to check yourself.

Submission instructions

1. Send your results via email to crypto-netsec@eng.tau.ac.il.

2. The subject should be: ex2. Do NOT put a dash (“-”) between the “x” and the “2” as it confuses the mailer.

3. The body of the email should contain 3 lines, including the leading keywords and the “:=” symbols:

   \[
   \begin{align*}
   TZ & := \text{your "Teudat Zehut" number (9 digits)} \\
   MIN & := \text{the minimal Hamming distance from question 2.} \\
   MAX & := \text{the maximal Hamming distance from question 2.}
   \end{align*}
   \]

4. Send plain ASCII email. In particular:

   (a) No attachments

   (b) No HTML email: configure your mail program to generate “unformatted” or “Plain text” messages only (not “Rich formatting”)

   (c) Be extra careful with Outlook which by default sends the text encapsulated in an attachment called “winmail.dat”. A recipe on how to fix this can be found in http://facstaff.gpc.edu/~jbenson/resource/winmail.htm (scroll all the way down)

   (d) When in doubt, use a Unix text-based mailer like “mail” or “pine”.