1. **Programming** Generate your own 31- or 32-bit RSA public and private keys:

   - Pick 2 secret primes $p, q$, in the range $[10000-65535]$: Try a few numbers at random and test for primality. You will find a prime very quickly. You can test a number $x$ for primality by checking whether all the odd numbers $y < \sqrt{x}$ do not divide $x$.
   
   - Compute your **public modulus** $n_z = pq$.
   
   - Compute the secret $\phi_z = (p - 1)(q - 1)$.
   
   - Pick a value for your **public exponent** $e_z$ such that $gcd(\phi_z, e_z) = 1$. Any value that works is OK.
   
   - Compute the secret exponent $d_z = (e_z)^{-1} \pmod {\phi_z}$ using the extended Euclid algorithm.
   
   - Tip: Test your key pair $(n_z, e_z)$: compute $t = 2^{e_z} \pmod {n_z}$ and check whether $t^{d_z} \pmod {n_z} = 2$.
   
   - Tip: Use unsigned 64-bit variables in your program to avoid integer overflows in intermediate values.
   
   - Tip: Make sure to pick $p$ and $q$ so $n$ is larger than your TZ number

Once you have a working RSA system $(n_z, e_z, d_z)$, **sign** your Teudat Zehut number (seen as a 9-digit integer): that is, compute

$$SIG = (TZ)^{d_z} \pmod {n_z}.$$ 

Send the results by email according to the instructions below.

2. Consider an RSA system with a public key $(n, e)$, with $n = pq$.

   (a) Show an algorithm to find the factors of $n$ if the secret $\phi(n)$ is leaked. Hint: write and solve a quadratic equation. Justify why your solution works on integers.

   (b) What is the time complexity of your algorithm (order of magnitude as a function of $n$)? Take into account the complexity of computing integer square roots.

3. (a) What is the probability that a randomly chosen number $a$ would not be relatively prime to a given RSA modulus $n$?

   (b) Estimate this probability in terms of $n$ for $p \approx q$ and for $p \approx q^3$.

   (c) What threat would such a number $a$ pose?

4. In a given RSA system with public $(n, e)$, prove that there exist more than one possible secret exponent $d$ that works. In other words, show that there exists $d' < \phi(n)$, $d' \neq e^{-1} \pmod {\phi(n)}$, which correctly decrypts every message $C = m^e \pmod n$. 
Submission Instructions

1. Hand in questions 2-4 on paper.

2. Send question 1 via email to

   crypto-netsec@eng.tau.ac.il

3. The subject should be: ex5. Do NOT put a dash ("-"") between the "x" and the "5" as it confuses the mailer.

4. The body of the email should contain 4 lines, including the leading keywords and the "::=" symbols:

   TZ := your "Teudat Zehut" number (9 digits)
   NZ := the public modulus \( n_z \) from q.1
   EZ := the public exponent \( e_z \) from q.1
   SIG := the signature from q.2, \((TZ)^{d_z} \mod n_z\) from q.1.

   Note that these are all PUBLIC values. Do NOT send your secret keys!

5. Send plain ASCII email.