Tutorial session 2: Programming with Maxima
Dr. Daniel Duviol Tcheutia
October 7, 2017, Douala, Cameroon

Recall:
(a) The `for` loop provides the ability to execute a statement repeatedly for a fixed number of times.
(b) If we want to repeat a statement, but we do not know how many times, we can use the `while` loop. The expression between `while` and `do` has to be a boolean expression (an expression returning `true` or `false`).
(c) Sometimes we want to make a case-by-case analysis. For this purpose we can use the `if` statement. The statements after `then` are only evaluated, if the condition is true.

1. Define the list $L = [1, 2, \ldots, 1000]$.
   (a) Derive the list $L1 = [2^p + 1, p \in L]$.
   (b) Count the number of primes in $L1$.
   (c) Derive the list of all the prime elements in $L1$.
   (d) Derive the list $L2 = [p \in L \ / \ 2^p + 1 \text{ is prime}]$.
   (e) Check that each element in $L2$ can be written as power of 2.
   (f) Compute the arithmetic mean of the elements of L and return its numerical approximation.

2. Let $a, b \in \mathbb{N}_{\geq} 0$. The division with remainder gives the relation
$$a = b\,q + r,\ \ 0 \leq r < b. \tag{1}$$
   Use the fact that $\gcd(a, b) = \gcd(b, r)$ and $\gcd(a, 0) = a$ to implement the function `gcd1(a,b)` which computes recursively the greatest common divisor of a and b. Hint: The remainder r in (1) is `mod(a,b)` in Maxima.

   Test your program for the computation of $\gcd(2^{400} + 3; 3^{300} + 8)$ and compare the timings with the internal function `gcd`.

3. Consider the matrix
$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 5 \\ 7 & 8 & 9 \end{pmatrix}.$$
   (a) Write a first recursive program `MatPow1(A,n)` to compute A^n $(n = 0, 1, 2, \ldots)$ using the relation $A^n = A \times A^{n-1}$. Apply this with M^{10} and M^{1000} (use `.` for matrix multiplication and `ident(n)` to get the identity matrix of order n).
   (b) Write the second recursive program `MatPow2(A,n)` to compute A^n $(n = 0, 1, 2, \ldots)$ using the divide-and-conquer approach: $A^n = (A^2)^{n/2}$ if n is even and $A^n = A \times A^{n-1}$ if n is odd.

   For each of the implementations, compare the timings with the internal function `A^^n` for $n = 10000$ for example.

4. Implement the procedures `PolyQuot(a, b, x)` and `PolyRem(a, b, x)` which compute, respectively, the quotient and the remainder of $a(x)$ by $b(x)$ for two polynomials $a(x)$, $b(x)$. Use your function to find the polynomial quotient and remainder of the division of $a(x) = 12x^6 - 8x^5 + 17x^4 + 5x^3 - 4x^2 + 3x + 5$ by $b(x) = 3x^4 + 5x - 1$ and $b(x)$ by $a(x)$ and check your results with the internal command `divide(a, b)`.
   Hint: `coeff(p,x,n)` returns the coefficient of x^n in the polynomial p, `hipow(p,x)` returns the degree of the polynomial p.

5. The Chebyshev polynomials $T_n(x)$ are defined by

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x),\ T_0(x) = 1 \text{ and } T_1(x) = x; \tag{2}$$

and have the property

$$\begin{cases} T_n(x) = 2\left(T_{\frac{n}{2}}(x)\right)^2 - 1, \text{ if } n \text{ is even} \\ T_n(x) = 2T_{\frac{n-1}{2}}(x)T_{\frac{n+1}{2}}(x) - x, \text{ if } n \text{ is odd} \\ T_0(x) = 1,\ T_1(x) = x. \end{cases} \tag{3}$$

Implement each of the above recurrence equations and compare their complexities by taking higher values of n.