```
calcul := proc(n, x)
local s, c, t, i;
s := 1/2 (x - 1./x);
c := 1/2 (x + 1./x);
t := s/c;
if n ≥ 1 then
 for i from 1 to n do
  c := sqrt((1 + c)/2);
s := s/c;
t := s/c od fi;
c, s, t
end;
```

```
proc(n, x)                                                                    (1)
    local s, c, t, i;
    s := 1/2*x + -1/2*1./x;
    c := 1/2*x + 1/2*1./x;
    t := s/c;
    if 1 <= n then for i to n do c := sqrt(1/2 + 1/2*c); s := s/c; t := s/c end do end if;
    c, s, t
end proc
```

```
calcul(3, 2);
```

$$1.003755888,\ 0.6940147577,\ 0.6914178696 \qquad (2)$$

```
evalf(ln(2));
```

$$0.6931471806 \qquad (3)$$

```
approxi := proc(epsilon, x)
local n, l;
n := 1;
l := calcul(n, x);
while (l[2] - l[3]) > epsilon do
 n := n + 1;
 l := calcul(n, x) od;
l[2]
end;
```

```
proc(epsilon, x)                                                              (4)
    local n, l;
    n := 1;
    l := calcul(n, x);
```

```
    while epsilon < l[2] - l[3] do n := n + 1; l := calcul(n, x) end do;
    l[2]
end proc
```

```
approxi(10^(-6), 2);
```

0.6931473934 **(5)**

```
> with(linalg);
```

[BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix, subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian] **(6)**

```
>
```

```
comparaison := proc(x)
 local m, i, l;
 m := matrix(4, 5, 0);
 for i from 1 to 4 do
  l := calcul(i, x);
  m[i, 1] := l[2];
  m[i, 2] := 2/3 l[2] + 1/3 l[3] od;
 for i from 1 to 3 do
  m[i, 3] := 4/3 m[i + 1, 1] - 1/3 m[i, 1] od;
 for i from 1 to 2 do
  m[i, 4] := 16/15 m[i + 1, 3] - 1/15 m[i, 3] od;
 m[1, 5] := 64/63 m[2, 4] - 1/63 m[1, 4];
 print(m);
 end;
```

```
proc(x)
    local m, i, l;
    m := linalg:-matrix(4, 5, 0);
```

(7)

```
    for i to 4 do l := calcul(i, x); m[i, 1] := l[2]; m[i, 2] := 2/3 * l[2] + 1/3 * l[3] end do;
    for i to 3 do m[i, 3] := 4/3 * m[1 + i, 1] − 1/3 * m[i, 1] end do;
    for i to 2 do m[i, 4] := 16/15 * m[1 + i, 3] − 1/15 * m[i, 3] end do;
    m[1, 5] := 64/63 * m[2, 4] − 1/63 * m[1, 4];
    print(m)
end proc
comparaison(2);
```

$$\begin{bmatrix} 0.7071067810 & 0.6936267428 & 0.6931262722 & 0.6931471842 & 0.6931471800 \\ 0.6966213994 & 0.6931781000 & 0.6931458771 & 0.6931471801 & 0 \\ 0.6940147577 & 0.6931491283 & 0.6931470986 & 0 & 0 \\ 0.6933640134 & 0.6931473020 & 0 & 0 & 0 \end{bmatrix} \quad \textbf{(8)}$$