testgrobner_sage-magma-giac

```
%sh
lscpu
```

```
/tmp/tmp0WpQzU
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                16
On-line CPU(s) list:   0-15
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 26
Stepping:              5
CPU MHz:               2261.017
BogoMIPS:              4521.83
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              8192K
NUMA node0 CPU(s):     0,2,4,6,8,10,12,14
NUMA node1 CPU(s):     1,3,5,7,9,11,13,15
```

```
A.<x,y>=PolynomialRing(ZZ, 2, order="degrevlex");
f  = -y^2 - y + x^3 + 7*x + 1;
fx=f.derivative(x);
fy=f.derivative(y);
II=ideal([f,fx,fy]);
```

```
from giacpy import *
```

NB: Giac groebner Basis are over QQ even if the answer have integer coefficients

```
II.groebner_basis(algorithm='macaulay2')
```

```
[x + 20607, y + 11314, 22627]
```

```
II.groebner_basis()
```

```
/home/han/dev/sage/local/lib/python2.7/site-packages/sage/libs/singu\
lar/standard_options.py:140:
**********************************************************************\
***********
Singular's groebner() and related computations in polynomial rings
over ZZ contains bugs and may be mathematically unreliable.
This issue is being tracked at
http://trac.sagemath.org/sage_trac/ticket/17676.
**********************************************************************\
***********
[x + 20607, y + 11314, 22627]
```

```
(libgiac(II.gens())).gbasis([A.gens()])
```

```
[1]
```

```
n=7;R=PolynomialRing(QQ,n,'x');
I = sage.rings.ideal.Cyclic(R,n);
I.gens()
```

```
[x0 + x1 + x2 + x3 + x4 + x5 + x6, x0*x1 + x1*x2 + x2*x3 + x3*x4 +
x4*x5 + x0*x6 + x5*x6, x0*x1*x2 + x1*x2*x3 + x2*x3*x4 + x3*x4*x5 +
x0*x1*x6 + x0*x5*x6 + x4*x5*x6, x0*x1*x2*x3 + x1*x2*x3*x4 +
x2*x3*x4*x5 + x0*x1*x2*x6 + x0*x1*x5*x6 + x0*x4*x5*x6 + x3*x4*x5*x6,
x0*x1*x2*x3*x4 + x1*x2*x3*x4*x5 + x0*x1*x2*x3*x6 + x0*x1*x2*x5*x6 +
x0*x1*x4*x5*x6 + x0*x3*x4*x5*x6 + x2*x3*x4*x5*x6, x0*x1*x2*x3*x4*x5
+ x0*x1*x2*x3*x4*x6 + x0*x1*x2*x3*x5*x6 + x0*x1*x2*x4*x5*x6 +
x0*x1*x3*x4*x5*x6 + x0*x2*x3*x4*x5*x6 + x1*x2*x3*x4*x5*x6,
x0*x1*x2*x3*x4*x5*x6 - 1]
```

```
#time B = I.groebner_basis(algorithm="libsingular:std")  #cancelled
```

```
Traceback (click to the left of this block for traceback)
...
__SAGE__
```

The CPU looks to be the time of conversion from magma to python??

```
#time B = I.groebner_basis(algorithm="magma",prot=True)
time B = I.groebner_basis(algorithm="magma")
```

```
Time: CPU 2.68 s, Wall: 4.77 s
```

Magma V2.20-10

```
%magma
R<x0,x1,x2,x3,x4,x5,x6> := PolynomialRing( RationalField(),7, "grevlex" );
L := [x0 + x1 + x2 + x3 + x4 + x5 + x6, x0*x1 + x1*x2 + x2*x3 + x3*x4 + x4*x5
+ x0*x6 + x5*x6, x0*x1*x2 + x1*x2*x3 + x2*x3*x4 + x3*x4*x5 + x0*x1*x6 +
x0*x5*x6 + x4*x5*x6, x0*x1*x2*x3 + x1*x2*x3*x4 + x2*x3*x4*x5 +
x0*x1*x2*x6 + x0*x1*x5*x6 + x0*x4*x5*x6 + x3*x4*x5*x6, x0*x1*x2*x3*x4 +
x1*x2*x3*x4*x5 + x0*x1*x2*x3*x6 + x0*x1*x2*x5*x6 + x0*x1*x4*x5*x6 +
x0*x3*x4*x5*x6 + x2*x3*x4*x5*x6, x0*x1*x2*x3*x4*x5 + x0*x1*x2*x3*x4*x6 +
x0*x1*x2*x3*x5*x6 + x0*x1*x2*x4*x5*x6 + x0*x1*x3*x4*x5*x6 +
x0*x2*x3*x4*x5*x6 + x1*x2*x3*x4*x5*x6, x0*x1*x2*x3*x4*x5*x6 - 1];
time B := GroebnerBasis(L);
```

```
Time: 1.020
```

```
giacsettings.threads=8
```

```
time BG=(libgiac(I.gens())).gbasis([R.gens()]);
```

```
    Running a probabilistic check for the reconstructed Groebner basis.
    If successfull, error probability is less than 1e-15 and is
    estimated to be less than 10^-91. Use proba_epsilon:=0 to certify
    (this takes more time).
    Time: CPU 6.13 s, Wall: 2.35 s
```

```
T=True;
for i in range(len(BG)):
    T=T and isinstance(((BG[i]/B[-i-1]).normal()).sage(),Integer)

if T:
    print("OK: The basis are proportional")
```

```
    OK: The basis are proportional
```

```
n=8;R=PolynomialRing(QQ,n,'x');
I = sage.rings.ideal.Cyclic(R,n);
I.gens()
```

```
    [x0 + x1 + x2 + x3 + x4 + x5 + x6 + x7, x0*x1 + x1*x2 + x2*x3 +
    x3*x4 + x4*x5 + x5*x6 + x0*x7 + x6*x7, x0*x1*x2 + x1*x2*x3 +
    x2*x3*x4 + x3*x4*x5 + x4*x5*x6 + x0*x1*x7 + x0*x6*x7 + x5*x6*x7,
    x0*x1*x2*x3 + x1*x2*x3*x4 + x2*x3*x4*x5 + x3*x4*x5*x6 + x0*x1*x2*x7
    + x0*x1*x6*x7 + x0*x5*x6*x7 + x4*x5*x6*x7, x0*x1*x2*x3*x4 +
    x1*x2*x3*x4*x5 + x2*x3*x4*x5*x6 + x0*x1*x2*x3*x7 + x0*x1*x2*x6*x7 +
    x0*x1*x5*x6*x7 + x0*x4*x5*x6*x7 + x3*x4*x5*x6*x7, x0*x1*x2*x3*x4*x5
    + x1*x2*x3*x4*x5*x6 + x0*x1*x2*x3*x4*x7 + x0*x1*x2*x3*x6*x7 +
    x0*x1*x2*x5*x6*x7 + x0*x1*x4*x5*x6*x7 + x0*x3*x4*x5*x6*x7 +
    x2*x3*x4*x5*x6*x7, x0*x1*x2*x3*x4*x5*x6 + x0*x1*x2*x3*x4*x5*x7 +
    x0*x1*x2*x3*x4*x6*x7 + x0*x1*x2*x3*x5*x6*x7 + x0*x1*x2*x4*x5*x6*x7 +
    x0*x1*x3*x4*x5*x6*x7 + x0*x2*x3*x4*x5*x6*x7 + x1*x2*x3*x4*x5*x6*x7,
    x0*x1*x2*x3*x4*x5*x6*x7 - 1]
```

NB: The CPU time looks like the time for converting from magma to python via pexpect.

```
time B = I.groebner_basis(algorithm="magma")
```

```
    Time: CPU 14.44 s, Wall: 47.50 s
```

```
%magma
P<x0,x1,x2,x3,x4,x5,x6,x7> := PolynomialRing(RationalField(),8,"grevlex");
I := Ideal([x0 + x1 + x2 + x3 + x4 + x5 + x6 + x7, x0*x1 + x1*x2 + x2*x3 + x3*x4 +
x4*x5 + x5*x6 + x0*x7 + x6*x7, x0*x1*x2 + x1*x2*x3 + x2*x3*x4 + x3*x4*x5
+ x4*x5*x6 + x0*x1*x7 + x0*x6*x7 + x5*x6*x7, x0*x1*x2*x3 + x1*x2*x3*x4 +
x2*x3*x4*x5 + x3*x4*x5*x6 + x0*x1*x2*x7 + x0*x1*x6*x7 + x0*x5*x6*x7 +
x4*x5*x6*x7, x0*x1*x2*x3*x4 + x1*x2*x3*x4*x5 + x2*x3*x4*x5*x6 +
x0*x1*x2*x3*x7 + x0*x1*x2*x6*x7 + x0*x1*x5*x6*x7 + x0*x4*x5*x6*x7 +
x3*x4*x5*x6*x7, x0*x1*x2*x3*x4*x5 + x1*x2*x3*x4*x5*x6 +
x0*x1*x2*x3*x4*x7 + x0*x1*x2*x3*x6*x7 + x0*x1*x2*x5*x6*x7 +
x0*x1*x4*x5*x6*x7 + x0*x3*x4*x5*x6*x7 + x2*x3*x4*x5*x6*x7,
x0*x1*x2*x3*x4*x5*x6 + x0*x1*x2*x3*x4*x5*x7 + x0*x1*x2*x3*x4*x6*x7 +
x0*x1*x2*x3*x5*x6*x7 + x0*x1*x2*x4*x5*x6*x7 + x0*x1*x3*x4*x5*x6*x7 +
x0*x2*x3*x4*x5*x6*x7 + x1*x2*x3*x4*x5*x6*x7, x0*x1*x2*x3*x4*x5*x6*x7 -
1]);
time B := GroebnerBasis(I);
```

```
    Time: 32.290
```

```
giacsettings.threads=8;
time BG=(libgiac(I.gens())).gbasis([R.gens()]);
```

```
    Running a probabilistic check for the reconstructed Groebner basis.
    If successfull, error probability is less than 1e-15 and is
    estimated to be less than 10^-105. Use proba_epsilon:=0 to certify
    (this takes more time).
    Time: CPU 204.64 s, Wall: 46.65 s
```

```
T=True;
for i in range(len(BG)):
    T=T and isinstance(((BG[i]/B[-i-1]).normal()).sage(),Integer)

if T:
    print("OK: The basis are proportional")
```

```
    OK: The basis are proportional
```

Trying some random examples

(6 equations of degree 3 with 6 variables)

```
n=6;R.<x0,x1,x2,x3,x4,x5>=PolynomialRing(QQ);
I=ideal([ R.random_element(degree=3) for i in range(n)]);
I.gens()
```

```
    [x0*x1*x4 + 6*x2^2*x5 + x2*x5 - 2*x4, 3*x1^2*x2 + x3^2*x4 + x1,
    -1/4*x4^3 + 2/5*x0*x1*x5 - 1/2*x1*x5^2, 1/2*x2*x4*x5 - x1^2 -
    9/2*x3*x4 + 1/3*x1 - 2, -6*x0*x2^2 + 1/38*x0*x1*x3 + 2*x1^2*x4 -
    8/7*x1*x4^2, 1/2*x1^3 + 2*x3^3 - 2*x1^2*x5 - 4*x3*x5^2 - 6*x1*x4]
```

```
time B = I.groebner_basis(algorithm="libsingular:std")
```

```
    Time: CPU 20.27 s, Wall: 20.31 s
```

```
%magma
P<x0,x1,x2,x3,x4,x5> := PolynomialRing(RationalField(),6,"grevlex");
I := Ideal([x2^2*x3 - 4/25*x0*x3^2 + x3^2*x4 - 14*x4^3, -1/3*x0*x3*x4 - 263*x3*x4^2
- x0^2*x5 + 3*x0*x2, -2*x0*x1*x5 + 1/2*x1*x2 + 1/11*x2^2, 3*x3^3 -
3*x0^2*x4 - 1/5*x0*x2*x5 + 4*x3*x5, 3*x3^3 - 1/3*x1*x3*x5 + 1/4*x0*x4*x5
+ 1/15*x3*x4*x5 + 2*x3*x4, -1/8*x0*x3*x4 + x3^2*x4 - 3/4*x1*x4^2 -
11*x1*x2*x5]);
time B := GroebnerBasis(I);
```

```
    Time: 2.510
```

```
L = [x2^2*x3 - 4/25*x0*x3^2 + x3^2*x4 - 14*x4^3, -1/3*x0*x3*x4 - 263*x3*x4^2
- x0^2*x5 + 3*x0*x2, -2*x0*x1*x5 + 1/2*x1*x2 + 1/11*x2^2, 3*x3^3 -
```

```
3*x0^2*x4 - 1/5*x0*x2*x5 + 4*x3*x5, 3*x3^3 - 1/3*x1*x3*x5 + 1/4*x0*x4*x5
+ 1/15*x3*x4*x5 + 2*x3*x4, -1/8*x0*x3*x4 + x3^2*x4 - 3/4*x1*x4^2 -
11*x1*x2*x5];
time BG= libgiac(L).gbasis([R.gens()])
```

```
    Running a probabilistic check for the reconstructed Groebner basis.
    If successfull, error probability is less than 1e-07 and is
    estimated to be less than 10^-35. Use proba_epsilon:=0 to certify
    (this takes more time).
    Time: CPU 8.20 s, Wall: 3.06 s
```

```
%magma
P<x0,x1,x2,x3,x4,x5> := PolynomialRing(RationalField(),6,"grevlex");
I := Ideal([1/4*x1^3 - x0^2*x2 - x1^2*x5 + x0*x2 - 1/7*x4*x5, -x0*x1*x3 - x3^2*x4 +
x0*x1 + 35*x2*x3 + 6*x0, -1/6*x0*x1^2 + 7/2*x1^2*x3 + 5/4*x1*x3^2 -
5/11*x0*x4^2 - 1/3*x5, -x0*x3*x4 + 4/3*x2*x4*x5 - 1/11*x3*x5^2 -
1/2*x1*x5, -2*x3*x4^2 - 2*x0*x2*x5 - 1/51*x1*x5^2 - 1/3*x4, 2/9*x2^3 -
x1*x3*x5 - 10*x4^2*x5 - 1/3*x4*x5^2 - 5*x0*x4]);
time B := GroebnerBasis(I);
```

```
    Time: 95.640
```

```
giacsettings.threads=8;
L = [1/4*x1^3 - x0^2*x2 - x1^2*x5 + x0*x2 - 1/7*x4*x5, -x0*x1*x3 - x3^2*x4 +
x0*x1 + 35*x2*x3 + 6*x0, -1/6*x0*x1^2 + 7/2*x1^2*x3 + 5/4*x1*x3^2 -
5/11*x0*x4^2 - 1/3*x5, -x0*x3*x4 + 4/3*x2*x4*x5 - 1/11*x3*x5^2 -
1/2*x1*x5, -2*x3*x4^2 - 2*x0*x2*x5 - 1/51*x1*x5^2 - 1/3*x4, 2/9*x2^3 -
x1*x3*x5 - 10*x4^2*x5 - 1/3*x4*x5^2 - 5*x0*x4];
time BG= libgiac(L).gbasis([R.gens()])
```

```
    Running a probabilistic check for the reconstructed Groebner basis.
    If successfull, error probability is less than 1e-15 and is
    estimated to be less than 10^-49. Use proba_epsilon:=0 to certify
    (this takes more time).
    Time: CPU 118.43 s, Wall: 49.36 s
```

```
B=(ideal(L)).groebner_basis(algorithm='magma')
```

```
T=True;
for i in range(len(BG)):
    T=T and isinstance(((BG[i]/B[-i-1]).normal()).sage(),Integer)

if T:
    print("OK: The basis are proportional")
```

```
    OK: The basis are proportional
```

```
n=5;R.<x0,x1,x2,x3,x4>=PolynomialRing(QQ);
I=ideal([ R.random_element(degree=4) for i in range(n)]);
I.gens()
```

```
    [1/4*x1*x2^3 + 10*x2^4 + 5/27*x1*x3^3 + 1/2*x2, 5*x2^2*x3^2 -
    1/5*x0^2*x2 - 14/5*x0*x1*x4 - x1*x4^2 - 1, 2*x0^2*x1*x2 -
    1/2*x2^3*x3 + 1/25*x1^2*x3^2 - 4*x3*x4^2, 3/349*x1*x2^2*x3 +
    8*x0*x1*x3*x4 + 4/3*x1*x2*x4^2 - 1/24*x3^2 + 4, -3/5*x0^2*x1*x3 +
    x0^2*x2*x4 + x0*x2*x4 - 12*x1*x2 - 1/207*x1*x3]
```

```
giacsettings.threads=8;
giacsettings.proba_epsilon=1e-7;
L = [1/4*x1*x2^3 + 10*x2^4 + 5/27*x1*x3^3 + 1/2*x2, 5*x2^2*x3^2 -
1/5*x0^2*x2 - 14/5*x0*x1*x4 - x1*x4^2 - 1, 2*x0^2*x1*x2 - 1/2*x2^3*x3 +
1/25*x1^2*x3^2 - 4*x3*x4^2, 3/349*x1*x2^2*x3 + 8*x0*x1*x3*x4 +
4/3*x1*x2*x4^2 - 1/24*x3^2 + 4, -3/5*x0^2*x1*x3 + x0^2*x2*x4 + x0*x2*x4
- 12*x1*x2 - 1/207*x1*x3];
time BG= libgiac(L).gbasis([R.gens()])
```

```
    Running a probabilistic check for the reconstructed Groebner basis.
    If successfull, error probability is less than 1e-07 and is
    estimated to be less than 10^-49. Use proba_epsilon:=0 to certify
    (this takes more time).
    Time: CPU 32.36 s, Wall: 13.74 s
```

```
%magma
P<x0,x1,x2,x3,x4> := PolynomialRing(RationalField(),5,"grevlex");
I := Ideal([1/4*x1*x2^3 + 10*x2^4 + 5/27*x1*x3^3 + 1/2*x2, 5*x2^2*x3^2 -
1/5*x0^2*x2 - 14/5*x0*x1*x4 - x1*x4^2 - 1, 2*x0^2*x1*x2 - 1/2*x2^3*x3 +
1/25*x1^2*x3^2 - 4*x3*x4^2, 3/349*x1*x2^2*x3 + 8*x0*x1*x3*x4 +
4/3*x1*x2*x4^2 - 1/24*x3^2 + 4, -3/5*x0^2*x1*x3 + x0^2*x2*x4 + x0*x2*x4
- 12*x1*x2 - 1/207*x1*x3]);
time B := GroebnerBasis(I);
```

```
    Time: 9.080
```

```
n=8;R.<x0,x1,x2,x3,x4,x5,x6,x7>=PolynomialRing(QQ);
I=ideal([ R.random_element(degree=3) for i in range(n)]);
I.gens()
```

```
    [-1/2*x0*x2^2 + 4*x0*x2*x4 - 2*x1*x4 - 1/17*x3*x5, x1*x3^2 +
    x0*x2*x5 + 1/2*x3^2*x5 + x1*x2 + 2/5*x6, x0*x4^2 + 1, -11/7*x0*x3^2
    + x2^2*x4 + x2*x5^2 - 24*x1*x4*x6, x2^2*x3 - 8/3*x2*x3*x4 -
    1/3*x4^2*x6 + 2*x6*x7^2, -x0^2*x1 - 1/3*x1^2*x2 - 7/6*x1*x2*x6 -
    1/7*x0*x3*x6 - 1688/5*x5*x6*x7, -1/4*x0*x3*x5 + 1/19*x5^3,
    1/6*x0*x3*x4 + 9*x3^2*x5 - 2*x2*x4*x5]
```

```
giacsettings.threads=8;
giacsettings.proba_epsilon=1e-7;
L = [-1/2*x0*x2^2 + 4*x0*x2*x4 - 2*x1*x4 - 1/17*x3*x5, x1*x3^2 + x0*x2*x5 +
1/2*x3^2*x5 + x1*x2 + 2/5*x6, x0*x4^2 + 1, -11/7*x0*x3^2 + x2^2*x4 +
x2*x5^2 - 24*x1*x4*x6, x2^2*x3 - 8/3*x2*x3*x4 - 1/3*x4^2*x6 + 2*x6*x7^2,
-x0^2*x1 - 1/3*x1^2*x2 - 7/6*x1*x2*x6 - 1/7*x0*x3*x6 - 1688/5*x5*x6*x7,
-1/4*x0*x3*x5 + 1/19*x5^3, 1/6*x0*x3*x4 + 9*x3^2*x5 - 2*x2*x4*x5];
time BG= libgiac(L).gbasis([R.gens()])
```

```
    Running a probabilistic check for the reconstructed Groebner basis.
    If successfull, error probability is less than 1e-07 and is
    estimated to be less than 10^-21. Use proba_epsilon:=0 to certify
    (this takes more time).
    Time: CPU 1483.94 s, Wall: 525.82 s
```

```
%magma
```

```
P<x0,x1,x2,x3,x4,x5,x6,x7> := PolynomialRing(RationalField(),8,"grevlex");
I := Ideal([-1/2*x0*x2^2 + 4*x0*x2*x4 - 2*x1*x4 - 1/17*x3*x5, x1*x3^2 + x0*x2*x5 +
1/2*x3^2*x5 + x1*x2 + 2/5*x6, x0*x4^2 + 1, -11/7*x0*x3^2 + x2^2*x4 +
x2*x5^2 - 24*x1*x4*x6, x2^2*x3 - 8/3*x2*x3*x4 - 1/3*x4^2*x6 + 2*x6*x7^2,
-x0^2*x1 - 1/3*x1^2*x2 - 7/6*x1*x2*x6 - 1/7*x0*x3*x6 - 1688/5*x5*x6*x7,
-1/4*x0*x3*x5 + 1/19*x5^3, 1/6*x0*x3*x4 + 9*x3^2*x5 - 2*x2*x4*x5]);
time B := GroebnerBasis(I);
```

```
Time: 2054.720
```

Test for save/load the (big) result with the giac savegen/loadgiacgen

NB: The savegen method is better than a string save because the string representation of an expression does not have the tree structure of a gen.

```
time BG.savegen('/home/han/testgrobnerdim8cubics.giacgen');
```

```
Time: CPU 7.47 s, Wall: 8.43 s
```

```
%sh
ls -hl /home/han/testgrobnerdim8cubics.giacgen
```

```
/tmp/tmpXLUW84
-rw-r--r-- 1 han 17000 350M juin   3 12:04
/home/han/testgrobnerdim8cubics.giacgen
```

```
time BGbis=loadgiacgen('/home/han/testgrobnerdim8cubics.giacgen')
```

```
Time: CPU 19.36 s, Wall: 19.39 s
```

```
(BG-BGbis).normal() # check the difference after save/load
```

```
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\
,0,0,0,0]
```