Claim: <.,.> is non-degenerate on Z. That is, if z in Z, and

<z, zstar> = 0

for all zstar in Z, then z=0.

```
ConstraintsZ = {};

(*
 * Z and Zstar are antisymmetric in upper and lower indices.
 * Let us therefore always represent them in normal forms.
 *)
ZN[a_, b_, i_, j_] :=
 Signature[{a, b}] Signature[{i, j}] z[Min[{a, b}], Max[{a, b}], Min[{i, j}], Max[{i, j}]]
ZstarN[a_, b_, i_, j_] := Signature[{a, b}] Signature[{i, j}]
  zstar[Min[{a, b}], Max[{a, b}], Min[{i, j}], Max[{i, j}]]
```

- **Notation:**

  ZN[i,j,k,l] = valid for all indices i,j,k,l
  z[i,j,k,l] = only defined for i<j and k<l

  ZstarN[i,j,k,l] and zstar are related similarly.

  Above ij are upper indices, and kl are lower indices.

- **In this notebook, trace means the first trace where only the latter index is contracted.**

- **Define modules for later use:**

```
(*
  Zero out all components of zstar[i,j,k,l].
 *)
ZeroZstar[] :=
  Module[{i, j, a, b},
   For[i = 1, i ≤ 4, i++,
    For[j = i + 1, j ≤ 4, j++,
      For[a = 1, a ≤ 4, a++,
        For[b = a + 1, b ≤ 4, b++,
          zstar[i, j, a, b] = 0;
         ];
       ];
     ];
   ];
 ]

(*
  Compute <Z,Zstar>.
   Note that z[i,j,k,l] and zstar[i,j,k,l] are global variables.
   The result is a scalar.
 *)
ComputeInnerProduct[] :=
 Module[{},
  Simplify[Sum[
    ZN[p, q, l, m] ZstarN[l, m, p, q],
    {p, 1, 4}, {q, 1, 4}, {l, 1, 4}, {m, 1, 4}
   ],
   ConstraintsZ]
 ]

  (*
  * 1. check that Zstar has zero trace, so that Zstar is in subspace Z.
  * 2. Compute inner product <Z,Zstar> which by assumption is zero, and
  *    add this constraint to ConstraintsZ if it is not yet in the list.
  *
  *)
CheckZstar[] := Module[
   {temp, HasZeroTrace, InnProd, NewConstraint, i, j},
   temp = {};
```

```
 (* Check that Zstar has zero trace. There are four conditions. *)
 HasZeroTrace = True;
For[i = 1, i ≤ 4, i++,
 For[j = 1, j ≤ 4, j++,
   sum = Sum[ ZstarN[i, P, j, P] , {P, 1, 4}];
   temp = Append[temp, sum];
   If [Simplify[sum] == 0,
    (* trace zero, do nothing *),
    HasZeroTrace = False;
   ];
  ];
];

 If[HasZeroTrace == False,
  Print["Zstar does not have zero trace. Check input."];
  Print[temp];
  Return[];
 ];

Print["Zstar has zero trace -- OK"];

(* Check that inner product <Z,Zstar> is not zero *)
InnProd = ComputeInnerProduct[];
Print["<Z,Zstar>=", InnProd];

If[InnProd == 0,
 Print["Constraint is not new."];
 Return[];
];

(* We use Simplify[..] to factor away any leading constant
 *)
NewConstraint = Simplify[InnProd == 0];
Print["Adding constraint: ", NewConstraint];
Print[""];
ConstraintsZ = Append[ConstraintsZ, NewConstraint];
];
```

Start to build up constraints on Z

- **Let us first write out the constraints on Zstar so that we can find suitable Zstar to plug into <Z,Zstar>=0. With suitable Zstar this will give enough constraints on Z to deduce that Z=0.**

```
Constraints = {};
For[i = 1, i ≤ 4, i++,
 For[j = 1, j ≤ 4, j++,
  Constraints = Append[Constraints,
     0 ==
      Sum[
        ZstarN[i, P, j, P] , {P, 1, 4}]
    ];
 ]
]
Constraints // MatrixForm
```

$$\begin{pmatrix}
0 == \text{zstar}[1,2,1,2] + \text{zstar}[1,3,1,3] + \text{zstar}[1,4,1,4] \\
0 == \text{zstar}[1,3,2,3] + \text{zstar}[1,4,2,4] \\
0 == -\text{zstar}[1,2,2,3] + \text{zstar}[1,4,3,4] \\
0 == -\text{zstar}[1,2,2,4] - \text{zstar}[1,3,3,4] \\
0 == \text{zstar}[2,3,1,3] + \text{zstar}[2,4,1,4] \\
0 == \text{zstar}[1,2,1,2] + \text{zstar}[2,3,2,3] + \text{zstar}[2,4,2,4] \\
0 == \text{zstar}[1,2,1,3] + \text{zstar}[2,4,3,4] \\
0 == \text{zstar}[1,2,1,4] - \text{zstar}[2,3,3,4] \\
0 == -\text{zstar}[2,3,1,2] + \text{zstar}[3,4,1,4] \\
0 == \text{zstar}[1,3,1,2] + \text{zstar}[3,4,2,4] \\
0 == \text{zstar}[1,3,1,3] + \text{zstar}[2,3,2,3] + \text{zstar}[3,4,3,4] \\
0 == \text{zstar}[1,3,1,4] + \text{zstar}[2,3,2,4] \\
0 == -\text{zstar}[2,4,1,2] - \text{zstar}[3,4,1,3] \\
0 == \text{zstar}[1,4,1,2] - \text{zstar}[3,4,2,3] \\
0 == \text{zstar}[1,4,1,3] + \text{zstar}[2,4,2,3] \\
0 == \text{zstar}[1,4,1,4] + \text{zstar}[2,4,2,4] + \text{zstar}[3,4,3,4]
\end{pmatrix}$$

First apply 12 contraints that contain only two terms:

```
(* 0==zstar[1,3,2,3]+zstar[1,4,2,4] *)
ZeroZstar[];
zstar[1, 3, 2, 3] = 1; zstar[1, 4, 2, 4] = -1;
CheckZstar[]
(* 0==-zstar[1,2,2,3]+zstar[1,4,3,4] *)
ZeroZstar[];
zstar[1, 2, 2, 3] = 1; zstar[1, 4, 3, 4] = 1;
CheckZstar[]
(* 0==-zstar[1,2,2,4]-zstar[1,3,3,4] *)
ZeroZstar[];
zstar[1, 2, 2, 4] = 1; zstar[1, 3, 3, 4] = -1;
CheckZstar[]
(*0==zstar[2,3,1,3]+zstar[2,4,1,4]*)
ZeroZstar[];
zstar[2, 3, 1, 3] = 1; zstar[2, 4, 1, 4] = -1;
CheckZstar[]
(* 0==zstar[1,2,1,3]+zstar[2,4,3,4] *)
ZeroZstar[];
zstar[1, 2, 1, 3] = 1; zstar[2, 4, 3, 4] = -1;
CheckZstar[]
(*0==zstar[1,2,1,4]-zstar[2,3,3,4]*)
ZeroZstar[];
zstar[1, 2, 1, 4] = 1; zstar[2, 3, 3, 4] = 1;
CheckZstar[]
(*0==-zstar[2,3,1,2]+zstar[3,4,1,4]*)
ZeroZstar[];
zstar[2, 3, 1, 2] = 1; zstar[3, 4, 1, 4] = 1;
CheckZstar[]
(* 0==zstar[1,3,1,2]+zstar[3,4,2,4] *)
ZeroZstar[];
zstar[1, 3, 1, 2] = 1; zstar[3, 4, 2, 4] = -1;
CheckZstar[]
(* 0==zstar[1,3,1,4]+zstar[2,3,2,4] *)
ZeroZstar[];
zstar[1, 3, 1, 4] = 1; zstar[2, 3, 2, 4] = -1;
CheckZstar[]
(* 0==-zstar[2,4,1,2]-zstar[3,4,1,3] *)
ZeroZstar[];
zstar[2, 4, 1, 2] = 1; zstar[3, 4, 1, 3] = -1;
CheckZstar[]
(* 0==zstar[1,4,1,2]-zstar[3,4,2,3] *)
ZeroZstar[];
zstar[1, 4, 1, 2] = 1; zstar[3, 4, 2, 3] = 1;
CheckZstar[]
(* 0==zstar[1,4,1,3]+zstar[2,4,2,3]*)
ZeroZstar[];
zstar[1, 4, 1, 3] = 1; zstar[2, 4, 2, 3] = -1;
CheckZstar[]
```

```
Zstar has zero trace -- OK

<Z,Zstar>=4 (z[2, 3, 1, 3] - z[2, 4, 1, 4])

Adding constraint: z[2, 3, 1, 3] == z[2, 4, 1, 4]


Zstar has zero trace -- OK

<Z,Zstar>=4 (z[2, 3, 1, 2] + z[3, 4, 1, 4])

Adding constraint: z[2, 3, 1, 2] + z[3, 4, 1, 4] == 0


Zstar has zero trace -- OK
```

<Z,Zstar>=4 (z[2, 4, 1, 2] - z[3, 4, 1, 3])

Adding constraint: z[2, 4, 1, 2] == z[3, 4, 1, 3]

Zstar has zero trace -- OK

<Z,Zstar>=4 (z[1, 3, 2, 3] - z[1, 4, 2, 4])

Adding constraint: z[1, 3, 2, 3] == z[1, 4, 2, 4]

Zstar has zero trace -- OK

<Z,Zstar>=4 (z[1, 3, 1, 2] - z[3, 4, 2, 4])

Adding constraint: z[1, 3, 1, 2] == z[3, 4, 2, 4]

Zstar has zero trace -- OK

<Z,Zstar>=4 (z[1, 4, 1, 2] + z[3, 4, 2, 3])

Adding constraint: z[1, 4, 1, 2] + z[3, 4, 2, 3] == 0

Zstar has zero trace -- OK

<Z,Zstar>=4 (z[1, 2, 2, 3] + z[1, 4, 3, 4])

Adding constraint: z[1, 2, 2, 3] + z[1, 4, 3, 4] == 0

Zstar has zero trace -- OK

<Z,Zstar>=4 (z[1, 2, 1, 3] - z[2, 4, 3, 4])

Adding constraint: z[1, 2, 1, 3] == z[2, 4, 3, 4]

Zstar has zero trace -- OK

<Z,Zstar>=4 (z[1, 4, 1, 3] - z[2, 4, 2, 3])

Adding constraint: z[1, 4, 1, 3] == z[2, 4, 2, 3]

Zstar has zero trace -- OK

<Z,Zstar>=4 (z[1, 2, 2, 4] - z[1, 3, 3, 4])

Adding constraint: z[1, 2, 2, 4] == z[1, 3, 3, 4]

Zstar has zero trace -- OK

<Z,Zstar>=4 (z[1, 2, 1, 4] + z[2, 3, 3, 4])

Adding constraint: z[1, 2, 1, 4] + z[2, 3, 3, 4] == 0

Zstar has zero trace -- OK

<Z,Zstar>=4 (z[1, 3, 1, 4] - z[2, 3, 2, 4])

```
Adding constraint: z[1, 3, 1, 4] == z[2, 3, 2, 4]
```

- **Next we study the four constraints that are involve 3 terms. Note that these are not independent since they contain common variables.**
- **The four constraints are**

0 == zstar[1, 2, 1, 2] + zstar[1, 3, 1, 3] + zstar[1, 4, 1, 4]
0 == zstar[1, 2, 1, 2] + zstar[2, 3, 2, 3] + zstar[2, 4, 2, 4]
0 == zstar[1, 3, 1, 3] + zstar[2, 3, 2, 3] + zstar[3, 4, 3, 4]
0 == zstar[1, 4, 1, 4] + zstar[2, 4, 2, 4] + zstar[3, 4, 3, 4]

and they can be written as

0 = AA + BB + CC
0 = AA + DD + EE
0 = BB + DD + FF
0 = CC + EE + FF

```
ZeroZstar[];
zstar[1, 2, 1, 2] = AA; zstar[1, 3, 1, 3] = BB; zstar[1, 4, 1, 4] = CC;
zstar[2, 3, 2, 3] = DD; zstar[2, 4, 2, 4] = EE; zstar[3, 4, 3, 4] = FF;

Simplify[Sum[
  ZN[p, q, l, m] ZstarN[l, m, p, q],
  {p, 1, 4}, {q, 1, 4}, {l, 1, 4}, {m, 1, 4}
 ], ConstraintsZ]
```

```
4 (AA z[1, 2, 1, 2] + BB z[1, 3, 1, 3] +
   CC z[1, 4, 1, 4] + DD z[2, 3, 2, 3] + EE z[2, 4, 2, 4] + FF z[3, 4, 3, 4])
```

```
Solve[{
  AA + BB + CC == 0,
  AA + DD + EE == 0,
  BB + DD + FF == 0,
  CC + EE + FF == 0,
  BB - CC - DD + EE == 6,
  AA - CC - DD + FF == 0
 }, {AA, BB, CC, DD, EE, FF}]
```

```
{{AA → -1, BB → 2, CC → -1, DD → -1, EE → 2, FF → -1}}
```

```
Solve[{
  AA + BB + CC == 0,
  AA + DD + EE == 0,
  BB + DD + FF == 0,
  CC + EE + FF == 0,
  BB - CC - DD + EE == 0,
  AA - CC - DD + FF == 6
 }, {AA, BB, CC, DD, EE, FF}]
```

```
{{AA → 2, BB → -1, CC → -1, DD → -1, EE → -1, FF → 2}}
```

- **Based on this we make two substitutions:**

```
ZeroZstar[];
zstar[1, 2, 1, 2] = -1; zstar[1, 3, 1, 3] = 2; zstar[1, 4, 1, 4] = -1;
zstar[2, 3, 2, 3] = -1; zstar[2, 4, 2, 4] = 2; zstar[3, 4, 3, 4] = -1;
CheckZstar[]

ZeroZstar[];
zstar[1, 2, 1, 2] = 2; zstar[1, 3, 1, 3] = -1; zstar[1, 4, 1, 4] = -1;
zstar[2, 3, 2, 3] = -1; zstar[2, 4, 2, 4] = -1; zstar[3, 4, 3, 4] = 2;
CheckZstar[]
```

```
Zstar has zero trace -- OK

<Z,Zstar>=
 -4 (z[1, 2, 1, 2] - 2 z[1, 3, 1, 3] + z[1, 4, 1, 4] + z[2, 3, 2, 3] - 2 z[2, 4, 2, 4] + z[3, 4, 3, 4])

Adding constraint:
 z[1, 2, 1, 2] + z[1, 4, 1, 4] + z[2, 3, 2, 3] + z[3, 4, 3, 4] == 2 (z[1, 3, 1, 3] + z[2, 4, 2, 4])


Zstar has zero trace -- OK

<Z,Zstar>=12 (z[1, 3, 1, 3] - z[1, 4, 1, 4] - z[2, 3, 2, 3] + z[2, 4, 2, 4])

Adding constraint: z[1, 3, 1, 3] + z[2, 4, 2, 4] == z[1, 4, 1, 4] + z[2, 3, 2, 3]
```

- **Next we make an observation: In trace Zstar[a,b,i,b] there are no terms where all indices are distinct. This gives (at most) 6 more constraints.**

```
PList = Permutations[{1, 2, 3, 4}];
For [perm = 1, perm ≤ Length[PList], perm++,
  ppp = PList[[perm]][[1]];
  qqq = PList[[perm]][[2]];
  aaa = PList[[perm]][[3]];
  bbb = PList[[perm]][[4]];

  If [ppp < qqq && aaa < bbb,
   Print["Assuming zstar[", ppp, ",", qqq, ",", aaa, ",", bbb, "]=1"];
   ZeroZstar[];
   zstar[ppp, qqq, aaa, bbb] = 1;
   CheckZstar[];
  ]
 ];
```

```
Assuming zstar[1,2,3,4]=1

Zstar has zero trace -- OK

<Z,Zstar>=4 z[3, 4, 1, 2]

Adding constraint: z[3, 4, 1, 2] == 0


Assuming zstar[1,3,2,4]=1

Zstar has zero trace -- OK

<Z,Zstar>=4 z[2, 4, 1, 3]

Adding constraint: z[2, 4, 1, 3] == 0


Assuming zstar[1,4,2,3]=1
```

Zstar has zero trace -- OK

<Z,Zstar>=4 z[2, 3, 1, 4]

Adding constraint: z[2, 3, 1, 4] == 0

Assuming zstar[2,3,1,4]=1

Zstar has zero trace -- OK

<Z,Zstar>=4 z[1, 4, 2, 3]

Adding constraint: z[1, 4, 2, 3] == 0

Assuming zstar[2,4,1,3]=1

Zstar has zero trace -- OK

<Z,Zstar>=4 z[1, 3, 2, 4]

Adding constraint: z[1, 3, 2, 4] == 0

Assuming zstar[3,4,1,2]=1

Zstar has zero trace -- OK

<Z,Zstar>=4 z[1, 2, 3, 4]

Adding constraint: z[1, 2, 3, 4] == 0

Assuming zstar[1,2,3,4]=1

Zstar has zero trace -- OK

<Z,Zstar>=4 z[3, 4, 1, 2]

Adding constraint: z[3, 4, 1, 2] == 0

Assuming zstar[1,3,2,4]=1

Zstar has zero trace -- OK

<Z,Zstar>=4 z[2, 4, 1, 3]

Adding constraint: z[2, 4, 1, 3] == 0

Assuming zstar[1,4,2,3]=1

Zstar has zero trace -- OK

<Z,Zstar>=4 z[2, 3, 1, 4]

Adding constraint: z[2, 3, 1, 4] == 0

Assuming zstar[2,3,1,4]=1

Zstar has zero trace -- OK

<Z,Zstar>=4 z[1, 4, 2, 3]

Adding constraint: z[1, 4, 2, 3] == 0

```
Assuming zstar[2,4,1,3]=1
Zstar has zero trace -- OK
<Z,Zstar>=4 z[1, 3, 2, 4]
Adding constraint: z[1, 3, 2, 4] == 0

Assuming zstar[3,4,1,2]=1
Zstar has zero trace -- OK
<Z,Zstar>=4 z[1, 2, 3, 4]
Adding constraint: z[1, 2, 3, 4] == 0

Assuming zstar[1,2,3,4]=1
Zstar has zero trace -- OK
<Z,Zstar>=4 z[3, 4, 1, 2]
Adding constraint: z[3, 4, 1, 2] == 0

Assuming zstar[1,3,2,4]=1
Zstar has zero trace -- OK
<Z,Zstar>=4 z[2, 4, 1, 3]
Adding constraint: z[2, 4, 1, 3] == 0

Assuming zstar[1,4,2,3]=1
Zstar has zero trace -- OK
<Z,Zstar>=4 z[2, 3, 1, 4]
Adding constraint: z[2, 3, 1, 4] == 0

Assuming zstar[2,3,1,4]=1
Zstar has zero trace -- OK
<Z,Zstar>=4 z[1, 4, 2, 3]
Adding constraint: z[1, 4, 2, 3] == 0

Assuming zstar[2,4,1,3]=1
Zstar has zero trace -- OK
<Z,Zstar>=4 z[1, 3, 2, 4]
Adding constraint: z[1, 3, 2, 4] == 0

Assuming zstar[3,4,1,2]=1
```

Zstar has zero trace -- OK

<Z,Zstar>=4 z[1, 2, 3, 4]

Adding constraint: z[1, 2, 3, 4] == 0

```
ConstraintsZ
Length[ConstraintsZ]
```

```
{z[2, 3, 1, 3] == z[2, 4, 1, 4], z[2, 3, 1, 2] + z[3, 4, 1, 4] == 0, z[2, 4, 1, 2] == z[3, 4, 1, 3],
 z[1, 3, 2, 3] == z[1, 4, 2, 4], z[1, 3, 1, 2] == z[3, 4, 2, 4], z[1, 4, 1, 2] + z[3, 4, 2, 3] == 0,
 z[1, 2, 2, 3] + z[1, 4, 3, 4] == 0, z[1, 2, 1, 3] == z[2, 4, 3, 4], z[1, 4, 1, 3] == z[2, 4, 2, 3],
 z[1, 2, 2, 4] == z[1, 3, 3, 4], z[1, 2, 1, 4] + z[2, 3, 3, 4] == 0, z[1, 3, 1, 4] == z[2, 3, 2, 4],
 z[1, 2, 1, 2] + z[1, 4, 1, 4] + z[2, 3, 2, 3] + z[3, 4, 3, 4] == 2 (z[1, 3, 1, 3] + z[2, 4, 2, 4]),
 z[1, 3, 1, 3] + z[2, 4, 2, 4] == z[1, 4, 1, 4] + z[2, 3, 2, 3], z[3, 4, 1, 2] == 0,
 z[2, 4, 1, 3] == 0, z[2, 3, 1, 4] == 0, z[1, 4, 2, 3] == 0, z[1, 3, 2, 4] == 0, z[1, 2, 3, 4] == 0}
```

20

- **Note: If z is in Zperp = { u in Omega^2_2 : <u,z>=0 for all z in Z} then z satisfies the above constraints. We will use this observation in notebook UZperpSubsetW.**

Last Step: We know that z belongs to Z, so add constraints for this.

- **Since Z has zero trace we obtain the following constraints:**

```
For[i = 1, i ≤ 4, i++,
 For[j = 1, j ≤ 4, j++,
  ConstraintsZ = Append[ConstraintsZ,
     0 == Sum[ ZN[i, P, j, P] , {P, 1, 4}]];
 ]
]
ConstraintsZ // MatrixForm
```

```
z[2, 3, 1, 3] == z[2, 4, 1, 4]
z[2, 3, 1, 2] + z[3, 4, 1, 4] == 0
z[2, 4, 1, 2] == z[3, 4, 1, 3]
z[1, 3, 2, 3] == z[1, 4, 2, 4]
z[1, 3, 1, 2] == z[3, 4, 2, 4]
z[1, 4, 1, 2] + z[3, 4, 2, 3] == 0
z[1, 2, 2, 3] + z[1, 4, 3, 4] == 0
z[1, 2, 1, 3] == z[2, 4, 3, 4]
z[1, 4, 1, 3] == z[2, 4, 2, 3]
z[1, 2, 2, 4] == z[1, 3, 3, 4]
z[1, 2, 1, 4] + z[2, 3, 3, 4] == 0
z[1, 3, 1, 4] == z[2, 3, 2, 4]
z[1, 2, 1, 2] + z[1, 4, 1, 4] + z[2, 3, 2, 3] + z[3, 4, 3, 4] == 2 (z[1, 3, 1, 3] + z[2, 4, 2, 4])
z[1, 3, 1, 3] + z[2, 4, 2, 4] == z[1, 4, 1, 4] + z[2, 3, 2, 3]
z[3, 4, 1, 2] == 0
z[2, 4, 1, 3] == 0
z[2, 3, 1, 4] == 0
z[1, 4, 2, 3] == 0
z[1, 3, 2, 4] == 0
z[1, 2, 3, 4] == 0
0 == z[1, 2, 1, 2] + z[1, 3, 1, 3] + z[1, 4, 1, 4]
0 == z[1, 3, 2, 3] + z[1, 4, 2, 4]
0 == -z[1, 2, 2, 3] + z[1, 4, 3, 4]
0 == -z[1, 2, 2, 4] - z[1, 3, 3, 4]
0 == z[2, 3, 1, 3] + z[2, 4, 1, 4]
0 == z[1, 2, 1, 2] + z[2, 3, 2, 3] + z[2, 4, 2, 4]
0 == z[1, 2, 1, 3] + z[2, 4, 3, 4]
0 == z[1, 2, 1, 4] - z[2, 3, 3, 4]
0 == -z[2, 3, 1, 2] + z[3, 4, 1, 4]
0 == z[1, 3, 1, 2] + z[3, 4, 2, 4]
0 == z[1, 3, 1, 3] + z[2, 3, 2, 3] + z[3, 4, 3, 4]
0 == z[1, 3, 1, 4] + z[2, 3, 2, 4]
0 == -z[2, 4, 1, 2] - z[3, 4, 1, 3]
0 == z[1, 4, 1, 2] - z[3, 4, 2, 3]
0 == z[1, 4, 1, 3] + z[2, 4, 2, 3]
0 == z[1, 4, 1, 4] + z[2, 4, 2, 4] + z[3, 4, 3, 4]
```

We now have enough constraints on z to show that z=0.

```
ZComponents = {};
For[i = 1, i ≤ 4, i++,
  For[j = 1, j ≤ 4, j++,
    For[a = 1, a ≤ 4, a++,
      For[b = 1, b ≤ 4, b++,
        ZComponents = Append[ZComponents,
            Simplify[ZN[i, j, a, b], ConstraintsZ]
          ];
      ];
    ];
  ];
 ];
ZComponents
```

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```