

NAG Toolbox for MATLAB

d06da

1 Purpose

d06da is a utility which performs an affine transformation of a given mesh.

2 Syntax

```
[coori, edgei, conni, cooro, edgeo, conno, ifail] = d06da(itype, trans,
coori, edgei, conni, itrace, 'nv', nv, 'nedge', nedge, 'nelt', nelt,
'ntrans', ntrans)
```

3 Description

d06da generates a mesh (co-ordinates, triangle/vertex connectivities and edge/vertex connectivities) resulting from an affine transformation of a given mesh. This transformation is of the form $Y = A \times X + B$, where

Y , X and B are in $\mathbb{R}^2$, and

A is a real 2 by 2 matrix.

Such a transformation includes a translation, a rotation, a scale reduction or increase, a symmetric transformation with respect to a user-supplied line, a user-supplied analytic transformation, or a composition of several transformations.

This function is partly derived from material in the MODULEF package from INRIA (Institut National de Recherche en Informatique et Automatique).

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **itype(ntrans)** – int32 array

itype(i), for $i = 1, \dots, \mathbf{ntrans}$, indicates the type of each transformation as follows:

itype(i) = 0

Identity transformation.

itype(i) = 1

Translation.

itype(i) = 2

Symmetric transformation with respect to a user-supplied line.

itype(i) = 3

Rotation.

itype(i) = 4

Scaling.

itype(i) = 10

User-supplied analytic transformation.

Note that the transformations are applied in the order described in **itype**.

Constraint: **itype**(i) = 0, 1, 2, 3, 4 or 10, for $i = 1, 2, \dots, \mathbf{ntrans}$.

2: **trans**(6,**ntrans**) – double array

The parameters for each transformation. For $i = 1, \dots, \mathbf{ntrans}$, **trans**(1, i) to **trans**(6, i) contain the parameters of the i th transformation.

If **itype**(i) = 0, elements **trans**(1, i) to **trans**(6, i) are not referenced.

If **itype**(i) = 1, the translation vector is $\vec{u} = \begin{pmatrix} a \\ b \end{pmatrix}$, where $a = \mathbf{trans}(1, i)$ and $b = \mathbf{trans}(2, i)$, while elements **trans**(3, i) to **trans**(6, i) are not referenced.

If **itype**(i) = 2, the user-supplied line is the curve $\{(x, y) \in \mathbb{R}^2; \text{ such that } ax + by + c = 0\}$, where $a = \mathbf{trans}(1, i)$, $b = \mathbf{trans}(2, i)$ and $c = \mathbf{trans}(3, i)$, while elements **trans**(4, i) to **trans**(6, i) are not referenced.

If **itype**(i) = 3, the centre of the rotation is (x_0, y_0) where $x_0 = \mathbf{trans}(1, i)$ and $y_0 = \mathbf{trans}(2, i)$, $\theta = \mathbf{trans}(3, i)$ is its angle in degrees, while elements **trans**(4, i) to **trans**(6, i) are not referenced.

If **itype**(i) = 4, $a = \mathbf{trans}(1, i)$ is the scaling coefficient in the x -direction, $b = \mathbf{trans}(2, i)$ is the scaling coefficient in the y -direction, and (x_0, y_0) are the scaling centre co-ordinates, with $x_0 = \mathbf{trans}(3, i)$ and $y_0 = \mathbf{trans}(4, i)$; while elements **trans**(5, i) to **trans**(6, i) are not referenced.

If **itype**(i) = 10, the user-supplied analytic affine transformation $Y = A \times X + B$ is such that $A = (a_{kl})_{1 \leq k, l \leq 2}$ and $B = (b_k)_{1 \leq k \leq 2}$ where $a_{kl} = \mathbf{trans}(2 \times (k - 1) + l, i)$, and $b_k = \mathbf{trans}(4 + k, i)$ with $k, l = 1, 2$.

3: **coori**(2,**nv**) – double array

coori(1, i) contains the x co-ordinate of the i th vertex of the input mesh, for $i = 1, \dots, \mathbf{nv}$; while **coori**(2, i) contains the corresponding y co-ordinate.

4: **edgei**(3,**nedge**) – int32 array

The specification of the boundary or interface edges. **edgei**(1, j) and **edgei**(2, j) contain the vertex numbers of the two end points of the j th boundary edge. **edgei**(3, j) is a user-supplied tag for the j th boundary edge.

Constraint: $1 \leq \mathbf{edgei}(i, j) \leq \mathbf{nv}$ and $\mathbf{edgei}(1, j) \neq \mathbf{edgei}(2, j)$, for $i = 1, 2$ and $j = 1, 2, \dots, \mathbf{nedge}$.

5: **conni(3,nelt) – int32 array**

The connectivity of the input mesh between triangles and vertices. For each triangle j , **conni**(i,j) gives the indices of its three vertices (in anticlockwise order), for $i = 1, 2, 3$ and $j = 1, \dots, \mathbf{nelt}$.

Constraints:

$$\begin{aligned} 1 &\leq \mathbf{conni}(i,j) \leq \mathbf{nv}; \\ \mathbf{conni}(1,j) &\neq \mathbf{conni}(2,j); \\ \mathbf{conni}(1,j) &\neq \mathbf{conni}(3,j) \text{ and } \mathbf{conni}(2,j) \neq \mathbf{conni}(3,j), \text{ for } i = 1, 2, 3 \text{ and } j = 1, 2, \dots, \mathbf{nelt}. \end{aligned}$$

6: **itrace – int32 scalar**

The level of trace information required from d06da.

$$\mathbf{itrace} \leq 0$$

No output is generated.

$$\mathbf{itrace} \geq 1$$

Details of each transformation, the matrix A and the vector B of the final transformation, which is the composition of all the **ntrans** transformations, are printed on the current advisory message unit (see x04ab).

5.2 Optional Input Parameters1: **nv – int32 scalar**

Default: The dimension of the arrays **coori**, **cooro**. (An error is raised if these dimensions are not equal.)

the total number of vertices in the input mesh.

Constraint: $\mathbf{nv} \geq 3$.

2: **nedge – int32 scalar**

Default: The dimension of the arrays **edgei**, **edgeo**. (An error is raised if these dimensions are not equal.)

the number of the boundary or interface edges in the input mesh.

Constraint: $\mathbf{nedge} \geq 1$.

3: **nelt – int32 scalar**

Default: The dimension of the arrays **conni**, **conno**. (An error is raised if these dimensions are not equal.)

the number of triangles in the input mesh.

Constraint: $\mathbf{nelt} \leq 2 \times \mathbf{nv} - 1$.

4: **ntrans – int32 scalar**

Default: The dimension of the arrays **itype**, **trans**. (An error is raised if these dimensions are not equal.)

the number of transformations of the input mesh.

Constraint: $\mathbf{ntrans} \geq 1$.

5.3 Input Parameters Omitted from the MATLAB Interface

rwork, lrwork

5.4 Output Parameters

1: **coori(2,nv)** – double array

see Section 8.

2: **edgei(3,nedge)** – int32 array

See Section 8.

3: **conni(3,nelt)** – int32 array

See Section 8.

4: **cooro(2,nv)** – double array

cooro(1,i) will contain the x co-ordinate of the i th vertex of the transformed mesh, for $i = 1, \dots, \mathbf{nv}$; while **cooro(2,i)** will contain the corresponding y co-ordinate.

5: **edgeo(3,nedge)** – int32 array

The specification of the boundary or interface edges of the transformed mesh. If the number of symmetric transformations is even or zero then **edgeo(1,j)** = **edgei(1,j)**, for $i = 1, 2, 3$ and $j = 1, \dots, \mathbf{nedge}$; otherwise **edgeo(1,j)** = **edgei(2,j)**, **edgeo(2,j)** = **edgei(1,j)** and **edgeo(3,j)** = **edgei(3,j)**, for $j = 1, \dots, \mathbf{nedge}$.

6: **conno(3,nelt)** – int32 array

The connectivity of the transformed mesh between triangles and vertices. If the number of symmetric transformations is even or zero then **conno(1,j)** = **conni(1,j)**, for $i = 1, 2, 3$ and $j = 1, \dots, \mathbf{nelt}$; otherwise **conno(1,j)** = **conni(1,j)**, **conno(2,j)** = **conni(3,j)** and **conno(3,j)** = **conni(2,j)**, for $j = 1, \dots, \mathbf{nelt}$.

7: **ifail** – int32 scalar

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **nv** < 3;
 or **nelt** > $2 \times \mathbf{nv} - 1$;
 or **nedge** < 1;
 or **edgei(1,j)** < 1 or **edgei(1,j)** > **nv** for some $i = 1, 2$ and $j = 1, \dots, \mathbf{nedge}$;
 or **edgei(2,j)** < 1 or **edgei(2,j)** > **nv** for some $j = 1, \dots, \mathbf{nedge}$;
 or **conni(1,j)** < 1 or **conni(1,j)** > **nv** for some $i = 1, 2, 3$ and $j = 1, \dots, \mathbf{nelt}$;
 or **conni(2,j)** < 1 or **conni(2,j)** > **nv** for some $j = 1, \dots, \mathbf{nelt}$;
 or **ntrans** < 1;
 or **itype(i)** $\neq 0, 1, 2, 3, 4$ or 10 for some $i = 1, \dots, \mathbf{ntrans}$;
 or **lrwork** < $12 \times \mathbf{ntrans}$.

ifail = 2

A serious error has occurred in an internal call to an auxiliary function. Check the input mesh especially the triangles/vertices and the edges/vertices connectivities as well as the details of each transformations.

7 Accuracy

Not applicable.

8 Further Comments

You may not wish to save the input mesh (**coori**, **edgei** and **conni**) and could call d06da using the same parameters for the input and the output (transformed) mesh.

9 Example

```
itype = [int32(1)];
trans = [0.7894736842105263;
        0.8947368421052632;
        0;
        0;
        0;
        0];
coori=zeros(2,400);
coori(1,1:400) = [0,
0.0526316,
0.105263,
0.157895,
0.210526,
0.263158,
0.315789,
0.368421,
0.421053,
0.473684,
0.526316,
0.578947,
0.631579,
0.684211,
0.736842,
0.789474,
0.842105,
0.894737,
0.947368,
1,
0,
0.053004,
0.106007,
0.15915,
0.209592,
0.260599,
0.313038,
0.366919,
0.420609,
0.47352,
0.5263370000000001,
0.579088,
0.631888,
0.684651,
0.737256,
0.78943,
0.841553,
0.894133,
0.947067,
```

```
1,  
0,  
0.0534333,  
0.106235,  
0.159956,  
0.209996,  
0.259977,  
0.311401,  
0.365802,  
0.420642,  
0.47341,  
0.526068,  
0.5786750000000001,  
0.631648,  
0.684596,  
0.737448,  
0.78982,  
0.84178,  
0.893995,  
0.946852,  
1,  
0,  
0.0538039,  
0.106545,  
0.158858,  
0.210128,  
0.260398,  
0.310943,  
0.3636,  
0.419451,  
0.472928,  
0.525577,  
0.578223,  
0.6310750000000001,  
0.683956,  
0.7368749999999999,  
0.78972,  
0.8420800000000001,  
0.894374,  
0.946955,  
1,  
0,  
0.05375149999999999,  
0.106776,  
0.158931,  
0.210208,  
0.260742,  
0.311521,  
0.363876,  
0.418145,  
0.47223,  
0.525111,  
0.577985,  
0.630736,  
0.68337,  
0.735968,  
0.788992,  
0.841819,  
0.894612,  
0.947184,  
1,  
0,  
0.0536725,  
0.106635,  
0.158876,  
0.210666,  
0.261283,  
0.312324,  
0.36466,  
0.418511,
```

```
0.471999,  
0.524845,  
0.577755,  
0.63064,  
0.683254,  
0.735891,  
0.788569,  
0.841356,  
0.894369,  
0.947225,  
1,  
0,  
0.054516,  
0.107159,  
0.158994,  
0.210434,  
0.261605,  
0.313007,  
0.365417,  
0.419192,  
0.473235,  
0.525468,  
0.577946,  
0.630796,  
0.683633,  
0.736328,  
0.78894,  
0.84111,  
0.894159,  
0.947129,  
1,  
0,  
0.056204,  
0.108298,  
0.159916,  
0.210824,  
0.261646,  
0.313118,  
0.365457,  
0.418953,  
0.473433,  
0.526137,  
0.57868,  
0.63134,  
0.684293,  
0.737208,  
0.78962,  
0.842011,  
0.8946,  
0.947268,  
1,  
0,  
0.0548594,  
0.108224,  
0.160449,  
0.211961,  
0.262781,  
0.313498,  
0.365485,  
0.418574,  
0.472328,  
0.525775,  
0.579114,  
0.631883,  
0.6846100000000001,  
0.737445,  
0.790017,  
0.842418,  
0.895069,  
0.947591,
```

```
1,  
0,  
0.0532319,  
0.106492,  
0.159501,  
0.211988,  
0.263778,  
0.315022,  
0.36643,  
0.418881,  
0.471646,  
0.52457,  
0.578628,  
0.63217,  
0.684783,  
0.736978,  
0.789629,  
0.842153,  
0.894842,  
0.94759,  
1,  
0,  
0.0514925,  
0.104211,  
0.157661,  
0.2111,  
0.26372,  
0.316379,  
0.368652,  
0.420519,  
0.472412,  
0.524455,  
0.57796,  
0.631985,  
0.684635,  
0.736406,  
0.788965,  
0.8417519999999999,  
0.894427,  
0.947283,  
1,  
0,  
0.0508223,  
0.10295,  
0.156261,  
0.209694,  
0.262775,  
0.316044,  
0.370283,  
0.423492,  
0.474535,  
0.526077,  
0.578169,  
0.631393,  
0.684214,  
0.736055,  
0.788503,  
0.841652,  
0.894472,  
0.947193,  
1,  
0,  
0.0515621,  
0.103835,  
0.156589,  
0.209397,  
0.262134,  
0.315575,  
0.370645,  
0.423719,
```

```
0.476017,  
0.52854,  
0.579371,  
0.6311639999999999,  
0.683326,  
0.7352919999999999,  
0.787892,  
0.841453,  
0.894691,  
0.947366,  
1,  
0,  
0.0526674,  
0.105418,  
0.158169,  
0.210509,  
0.262606,  
0.315494,  
0.369671,  
0.423157,  
0.475555,  
0.528315,  
0.5805709999999999,  
0.631576,  
0.682963,  
0.734669,  
0.78656,  
0.840591,  
0.894361,  
0.947369,  
1,  
0,  
0.0525525,  
0.105929,  
0.159138,  
0.211875,  
0.264131,  
0.315782,  
0.368611,  
0.421782,  
0.47416,  
0.527515,  
0.580596,  
0.632354,  
0.6833939999999999,  
0.73475,  
0.786534,  
0.839112,  
0.8934260000000001,  
0.9469959999999999,  
1,  
0,  
0.052283900000000001,  
0.105635,  
0.159138,  
0.21224,  
0.26492,  
0.316755,  
0.368556,  
0.421044,  
0.473635,  
0.526484,  
0.5801230000000001,  
0.633694,  
0.684453,  
0.73524,  
0.786532,  
0.838996,  
0.892616,  
0.946507,
```

```
1,  
0,  
0.0523473,  
0.105941,  
0.159189,  
0.212004,  
0.264979,  
0.317598,  
0.369415,  
0.421163,  
0.473189,  
0.525571,  
0.5790459999999999,  
0.632734,  
0.685273,  
0.7361839999999999,  
0.786892,  
0.838821,  
0.8921790000000001,  
0.9461540000000001,  
1,  
0,  
0.0519324,  
0.105695,  
0.159166,  
0.211854,  
0.264323,  
0.317522,  
0.369989,  
0.42178,  
0.473248,  
0.5252599999999999,  
0.578264,  
0.631788,  
0.6849460000000001,  
0.73676,  
0.7877420000000001,  
0.83928,  
0.892136,  
0.945949,  
1,  
0,  
0.0520265,  
0.105228,  
0.158898,  
0.21141,  
0.264014,  
0.316723,  
0.36949,  
0.421818,  
0.473756,  
0.5257039999999999,  
0.57829,  
0.631358,  
0.684441,  
0.736995,  
0.788488,  
0.840525,  
0.893076,  
0.946255,  
1,  
0,  
0.0526316,  
0.105263,  
0.157895,  
0.210526,  
0.263158,  
0.315789,  
0.368421,  
0.421053,
```



```
0.105263,  
0.157895,  
0.157947,  
0.158585,  
0.160703,  
0.161508,  
0.160854,  
0.159774,  
0.159143,  
0.159396,  
0.158225,  
0.157661,  
0.157998,  
0.158158,  
0.157847,  
0.157691,  
0.157682,  
0.157078,  
0.156562,  
0.156934,  
0.157895,  
0.210526,  
0.209851,  
0.209921,  
0.211672,  
0.214046,  
0.214868,  
0.214591,  
0.214038,  
0.212959,  
0.211629,  
0.210427,  
0.210469,  
0.21082,  
0.210792,  
0.210731,  
0.210732,  
0.210398,  
0.20989,  
0.209907,  
0.210526,  
0.263158,  
0.262201,  
0.261527,  
0.262359,  
0.264518,  
0.266476,  
0.267273,  
0.267047,  
0.265891,  
0.264455,  
0.263507,  
0.263408,  
0.263527,  
0.263258,  
0.262664,  
0.262596,  
0.262854,  
0.262874,  
0.262874,  
0.263158,  
0.315789,  
0.315767,  
0.314677,  
0.31412,  
0.314624,  
0.316212,  
0.317562,  
0.317948,  
0.317475,
```

```
0.316759,  
0.316165,  
0.316445,  
0.316505,  
0.316034,  
0.315338,  
0.315112,  
0.315404,  
0.3159,  
0.315842,  
0.315789,  
0.368421,  
0.36991,  
0.368859,  
0.367246,  
0.365954,  
0.365335,  
0.366449,  
0.367228,  
0.367797,  
0.368303,  
0.368453,  
0.369026,  
0.369213,  
0.368774,  
0.368247,  
0.368658,  
0.369427,  
0.369468,  
0.368954,  
0.368421,  
0.421053,  
0.423169,  
0.423363,  
0.421555,  
0.419389,  
0.417386,  
0.415919,  
0.416287,  
0.417343,  
0.418262,  
0.420205,  
0.421262,  
0.42146,  
0.421103,  
0.420855,  
0.421458,  
0.422405,  
0.422629,  
0.421965,  
0.421053,  
0.473684,  
0.475673,  
0.477027,  
0.476105,  
0.474016,  
0.471957,  
0.470082,  
0.46891,  
0.469108,  
0.46938,  
0.471703,  
0.473569,  
0.473634,  
0.473058,  
0.472478,  
0.472845,  
0.473917,  
0.474621,  
0.474445,
```

```
0.473684,  
0.526316,  
0.527735,  
0.529574,  
0.52977,  
0.528675,  
0.527155,  
0.526398,  
0.525467,  
0.525018,  
0.525035,  
0.5255570000000001,  
0.526664,  
0.52644,  
0.525219,  
0.524028,  
0.523759,  
0.524678,  
0.525944,  
0.526498,  
0.526316,  
0.578947,  
0.580673,  
0.582013,  
0.582639,  
0.582137,  
0.5816210000000001,  
0.582413,  
0.581142,  
0.582131,  
0.581193,  
0.580824,  
0.580756,  
0.580005,  
0.578309,  
0.57646,  
0.575477,  
0.576006,  
0.577533,  
0.578628,  
0.578947,  
0.631579,  
0.633678,  
0.634871,  
0.635456,  
0.635196,  
0.63472,  
0.634618,  
0.634457,  
0.633784,  
0.633788,  
0.634864,  
0.6341639999999999,  
0.633554,  
0.631992,  
0.6295770000000001,  
0.628376,  
0.628291,  
0.629566,  
0.6309129999999999,  
0.631579,  
0.684211,  
0.686277,  
0.687895,  
0.688395,  
0.6880039999999999,  
0.6873089999999999,  
0.686923,  
0.686428,  
0.685926,
```

```
0.685169,  
0.685377,  
0.685916,  
0.6862279999999999,  
0.685657,  
0.684106,  
0.682127,  
0.681347,  
0.681853,  
0.683167,  
0.684211,  
0.736842,  
0.738402,  
0.740228,  
0.740783,  
0.740282,  
0.739646,  
0.739109,  
0.739363,  
0.738853,  
0.737399,  
0.736712,  
0.737286,  
0.738184,  
0.73854,  
0.738167,  
0.73676,  
0.734838,  
0.734706,  
0.735569,  
0.736842,  
0.789474,  
0.790785,  
0.791683,  
0.791887,  
0.791595,  
0.790667,  
0.790307,  
0.791156,  
0.791569,  
0.790315,  
0.78866,  
0.78845,  
0.789734,  
0.790438,  
0.791083,  
0.7907960000000001,  
0.789559,  
0.788465,  
0.788501,  
0.789474,  
0.842105,  
0.8430800000000001,  
0.842714,  
0.842502,  
0.842294,  
0.841753,  
0.841354,  
0.842226,  
0.843557,  
0.843214,  
0.841287,  
0.840113,  
0.839827,  
0.841863,  
0.843417,  
0.843934,  
0.843293,  
0.842328,  
0.841808,
```

[illegible]

```
int32(9),
int32(10),
int32(11),
int32(12),
int32(13),
int32(14),
int32(15),
int32(16),
int32(17),
int32(18),
int32(19),
int32(20),
int32(40),
int32(60),
int32(80),
int32(100),
int32(120),
int32(140),
int32(160),
int32(180),
int32(200),
int32(220),
int32(240),
int32(260),
int32(280),
int32(300),
int32(320),
int32(340),
int32(360),
int32(380),
int32(400),
int32(399),
int32(398),
int32(397),
int32(396),
int32(395),
int32(394),
int32(393),
int32(392),
int32(391),
int32(390),
int32(389),
int32(388),
int32(387),
int32(386),
int32(385),
int32(384),
int32(383),
int32(382),
int32(381),
int32(361),
int32(341),
int32(321),
int32(301),
int32(281),
int32(261),
int32(241),
int32(221),
int32(201),
int32(181),
int32(161),
int32(141),
int32(121),
int32(101),
int32(81),
int32(61),
int32(41),
int32(21)];
edgei(2,1:76) = [ int32(2),
int32(3),
```

```
int32(4),  
int32(5),  
int32(6),  
int32(7),  
int32(8),  
int32(9),  
int32(10),  
int32(11),  
int32(12),  
int32(13),  
int32(14),  
int32(15),  
int32(16),  
int32(17),  
int32(18),  
int32(19),  
int32(20),  
int32(40),  
int32(60),  
int32(80),  
int32(100),  
int32(120),  
int32(140),  
int32(160),  
int32(180),  
int32(200),  
int32(220),  
int32(240),  
int32(260),  
int32(280),  
int32(300),  
int32(320),  
int32(340),  
int32(360),  
int32(380),  
int32(400),  
int32(399),  
int32(398),  
int32(397),  
int32(396),  
int32(395),  
int32(394),  
int32(393),  
int32(392),  
int32(391),  
int32(390),  
int32(389),  
int32(388),  
int32(387),  
int32(386),  
int32(385),  
int32(384),  
int32(383),  
int32(382),  
int32(381),  
int32(361),  
int32(341),  
int32(321),  
int32(301),  
int32(281),  
int32(261),  
int32(241),  
int32(221),  
int32(201),  
int32(181),  
int32(161),  
int32(141),  
int32(121),  
int32(101),  
int32(81),
```



```
int32(1),  
int32(1),  
int32(1),  
int32(1),  
int32(1),  
int32(1),  
int32(1),  
int32(1),  
int32(1),  
int32(1)];  
conni=zeros(3,722, 'int32');  
conni(1,1:722) = [int32(1),  
int32(1),  
int32(2),  
int32(2),  
int32(3),  
int32(3),  
int32(4),  
int32(4),  
int32(5),  
int32(5),  
int32(6),  
int32(6),  
int32(7),  
int32(7),  
int32(8),  
int32(8),  
int32(9),  
int32(9),  
int32(10),  
int32(10),  
int32(11),  
int32(11),  
int32(12),  
int32(12),  
int32(13),  
int32(13),  
int32(14),  
int32(14),  
int32(15),  
int32(15),  
int32(16),  
int32(16),  
int32(17),  
int32(17),  
int32(18),  
int32(18),  
int32(19),  
int32(19),  
int32(21),  
int32(21),  
int32(22),  
int32(22),  
int32(23),  
int32(23),  
int32(24),  
int32(24),  
int32(25),  
int32(25),  
int32(26),  
int32(26),  
int32(27),  
int32(27),  
int32(28),  
int32(28),  
int32(29),  
int32(29),  
int32(30),  
int32(30),  
int32(31),
```

```
int32(31),
int32(32),
int32(32),
int32(33),
int32(33),
int32(34),
int32(34),
int32(35),
int32(35),
int32(36),
int32(36),
int32(37),
int32(37),
int32(38),
int32(38),
int32(39),
int32(39),
int32(41),
int32(41),
int32(42),
int32(42),
int32(43),
int32(43),
int32(44),
int32(44),
int32(45),
int32(45),
int32(46),
int32(46),
int32(47),
int32(47),
int32(48),
int32(48),
int32(49),
int32(49),
int32(50),
int32(50),
int32(51),
int32(51),
int32(52),
int32(52),
int32(53),
int32(53),
int32(54),
int32(54),
int32(55),
int32(55),
int32(56),
int32(56),
int32(57),
int32(57),
int32(58),
int32(58),
int32(59),
int32(59),
int32(61),
int32(61),
int32(62),
int32(62),
int32(63),
int32(63),
int32(64),
int32(64),
int32(65),
int32(65),
int32(66),
int32(66),
int32(67),
int32(67),
int32(68),
```

```
int32(68),  
int32(69),  
int32(69),  
int32(70),  
int32(70),  
int32(71),  
int32(71),  
int32(72),  
int32(72),  
int32(73),  
int32(73),  
int32(74),  
int32(74),  
int32(75),  
int32(75),  
int32(76),  
int32(76),  
int32(77),  
int32(77),  
int32(78),  
int32(78),  
int32(79),  
int32(79),  
int32(81),  
int32(81),  
int32(82),  
int32(82),  
int32(83),  
int32(83),  
int32(84),  
int32(84),  
int32(85),  
int32(85),  
int32(86),  
int32(86),  
int32(87),  
int32(87),  
int32(88),  
int32(88),  
int32(89),  
int32(89),  
int32(90),  
int32(90),  
int32(91),  
int32(91),  
int32(92),  
int32(92),  
int32(93),  
int32(93),  
int32(94),  
int32(94),  
int32(95),  
int32(95),  
int32(96),  
int32(96),  
int32(97),  
int32(97),  
int32(98),  
int32(98),  
int32(99),  
int32(99),  
int32(101),  
int32(101),  
int32(102),  
int32(102),  
int32(103),  
int32(103),  
int32(104),  
int32(104),  
int32(105),
```

```
int32(105),  
int32(106),  
int32(106),  
int32(107),  
int32(107),  
int32(108),  
int32(108),  
int32(109),  
int32(109),  
int32(110),  
int32(110),  
int32(111),  
int32(111),  
int32(112),  
int32(112),  
int32(113),  
int32(113),  
int32(114),  
int32(114),  
int32(115),  
int32(115),  
int32(116),  
int32(116),  
int32(117),  
int32(117),  
int32(118),  
int32(118),  
int32(119),  
int32(119),  
int32(121),  
int32(121),  
int32(122),  
int32(122),  
int32(123),  
int32(123),  
int32(124),  
int32(124),  
int32(125),  
int32(125),  
int32(126),  
int32(126),  
int32(127),  
int32(127),  
int32(128),  
int32(128),  
int32(129),  
int32(129),  
int32(130),  
int32(130),  
int32(131),  
int32(131),  
int32(132),  
int32(132),  
int32(133),  
int32(133),  
int32(134),  
int32(134),  
int32(135),  
int32(135),  
int32(136),  
int32(136),  
int32(137),  
int32(137),  
int32(138),  
int32(138),  
int32(139),  
int32(139),  
int32(141),  
int32(141),  
int32(142),
```

```
int32(142),  
int32(143),  
int32(143),  
int32(144),  
int32(144),  
int32(145),  
int32(145),  
int32(146),  
int32(146),  
int32(147),  
int32(147),  
int32(148),  
int32(148),  
int32(149),  
int32(149),  
int32(150),  
int32(150),  
int32(151),  
int32(151),  
int32(152),  
int32(152),  
int32(153),  
int32(153),  
int32(154),  
int32(154),  
int32(155),  
int32(155),  
int32(156),  
int32(156),  
int32(157),  
int32(157),  
int32(158),  
int32(158),  
int32(159),  
int32(159),  
int32(161),  
int32(161),  
int32(162),  
int32(162),  
int32(163),  
int32(163),  
int32(164),  
int32(164),  
int32(165),  
int32(165),  
int32(166),  
int32(166),  
int32(167),  
int32(167),  
int32(168),  
int32(168),  
int32(169),  
int32(169),  
int32(170),  
int32(170),  
int32(171),  
int32(171),  
int32(172),  
int32(172),  
int32(173),  
int32(173),  
int32(174),  
int32(174),  
int32(175),  
int32(175),  
int32(176),  
int32(176),  
int32(177),  
int32(177),  
int32(178),
```

```
int32(178),  
int32(179),  
int32(179),  
int32(181),  
int32(181),  
int32(182),  
int32(182),  
int32(183),  
int32(183),  
int32(184),  
int32(184),  
int32(185),  
int32(185),  
int32(186),  
int32(186),  
int32(187),  
int32(187),  
int32(188),  
int32(188),  
int32(189),  
int32(189),  
int32(190),  
int32(190),  
int32(191),  
int32(191),  
int32(192),  
int32(192),  
int32(193),  
int32(193),  
int32(194),  
int32(194),  
int32(195),  
int32(195),  
int32(196),  
int32(196),  
int32(197),  
int32(197),  
int32(198),  
int32(198),  
int32(199),  
int32(199),  
int32(201),  
int32(201),  
int32(202),  
int32(202),  
int32(203),  
int32(203),  
int32(204),  
int32(204),  
int32(205),  
int32(205),  
int32(206),  
int32(206),  
int32(207),  
int32(207),  
int32(208),  
int32(208),  
int32(209),  
int32(209),  
int32(210),  
int32(210),  
int32(211),  
int32(211),  
int32(212),  
int32(212),  
int32(213),  
int32(213),  
int32(214),  
int32(214),  
int32(215),
```

```
int32(215),  
int32(216),  
int32(216),  
int32(217),  
int32(217),  
int32(218),  
int32(218),  
int32(219),  
int32(219),  
int32(221),  
int32(221),  
int32(222),  
int32(222),  
int32(223),  
int32(223),  
int32(224),  
int32(224),  
int32(225),  
int32(225),  
int32(226),  
int32(226),  
int32(227),  
int32(227),  
int32(228),  
int32(228),  
int32(229),  
int32(229),  
int32(230),  
int32(230),  
int32(231),  
int32(231),  
int32(232),  
int32(232),  
int32(233),  
int32(233),  
int32(234),  
int32(234),  
int32(235),  
int32(235),  
int32(236),  
int32(236),  
int32(237),  
int32(237),  
int32(238),  
int32(238),  
int32(239),  
int32(239),  
int32(241),  
int32(241),  
int32(242),  
int32(242),  
int32(243),  
int32(243),  
int32(244),  
int32(244),  
int32(245),  
int32(245),  
int32(246),  
int32(246),  
int32(247),  
int32(247),  
int32(248),  
int32(248),  
int32(249),  
int32(249),  
int32(250),  
int32(250),  
int32(251),  
int32(251),  
int32(252),
```

```
int32(252),  
int32(253),  
int32(253),  
int32(254),  
int32(254),  
int32(255),  
int32(255),  
int32(256),  
int32(256),  
int32(257),  
int32(257),  
int32(258),  
int32(258),  
int32(259),  
int32(259),  
int32(261),  
int32(261),  
int32(262),  
int32(262),  
int32(263),  
int32(263),  
int32(264),  
int32(264),  
int32(265),  
int32(265),  
int32(266),  
int32(266),  
int32(267),  
int32(267),  
int32(268),  
int32(268),  
int32(269),  
int32(269),  
int32(270),  
int32(270),  
int32(271),  
int32(271),  
int32(272),  
int32(272),  
int32(273),  
int32(273),  
int32(274),  
int32(274),  
int32(275),  
int32(275),  
int32(276),  
int32(276),  
int32(277),  
int32(277),  
int32(278),  
int32(278),  
int32(279),  
int32(279),  
int32(281),  
int32(281),  
int32(282),  
int32(282),  
int32(283),  
int32(283),  
int32(284),  
int32(284),  
int32(285),  
int32(285),  
int32(286),  
int32(286),  
int32(287),  
int32(287),  
int32(288),  
int32(288),  
int32(289),
```

```
int32(289),  
int32(290),  
int32(290),  
int32(291),  
int32(291),  
int32(292),  
int32(292),  
int32(293),  
int32(293),  
int32(294),  
int32(294),  
int32(295),  
int32(295),  
int32(296),  
int32(296),  
int32(297),  
int32(297),  
int32(298),  
int32(298),  
int32(299),  
int32(299),  
int32(301),  
int32(301),  
int32(302),  
int32(302),  
int32(303),  
int32(303),  
int32(304),  
int32(304),  
int32(305),  
int32(305),  
int32(306),  
int32(306),  
int32(307),  
int32(307),  
int32(308),  
int32(308),  
int32(309),  
int32(309),  
int32(310),  
int32(310),  
int32(311),  
int32(311),  
int32(312),  
int32(312),  
int32(313),  
int32(313),  
int32(314),  
int32(314),  
int32(315),  
int32(315),  
int32(316),  
int32(316),  
int32(317),  
int32(317),  
int32(318),  
int32(318),  
int32(319),  
int32(319),  
int32(321),  
int32(321),  
int32(322),  
int32(322),  
int32(323),  
int32(323),  
int32(324),  
int32(324),  
int32(325),  
int32(325),  
int32(326),
```

```
int32(326),
int32(327),
int32(327),
int32(328),
int32(328),
int32(329),
int32(329),
int32(330),
int32(330),
int32(331),
int32(331),
int32(332),
int32(332),
int32(333),
int32(333),
int32(334),
int32(334),
int32(335),
int32(335),
int32(336),
int32(336),
int32(337),
int32(337),
int32(338),
int32(338),
int32(339),
int32(339),
int32(341),
int32(341),
int32(342),
int32(342),
int32(343),
int32(343),
int32(344),
int32(344),
int32(345),
int32(345),
int32(346),
int32(346),
int32(347),
int32(347),
int32(348),
int32(348),
int32(349),
int32(349),
int32(350),
int32(350),
int32(351),
int32(351),
int32(352),
int32(352),
int32(353),
int32(353),
int32(354),
int32(354),
int32(355),
int32(355),
int32(356),
int32(356),
int32(357),
int32(357),
int32(358),
int32(358),
int32(359),
int32(359),
int32(361),
int32(361),
int32(362),
int32(362),
int32(363),
```

```
int32(363),
int32(364),
int32(364),
int32(365),
int32(365),
int32(366),
int32(366),
int32(367),
int32(367),
int32(368),
int32(368),
int32(369),
int32(369),
int32(370),
int32(370),
int32(371),
int32(371),
int32(372),
int32(372),
int32(373),
int32(373),
int32(374),
int32(374),
int32(375),
int32(375),
int32(376),
int32(376),
int32(377),
int32(377),
int32(378),
int32(378),
int32(379),
int32(379)];
conni(2,1:722) = [ int32(2),
int32(22),
int32(3),
int32(23),
int32(4),
int32(24),
int32(5),
int32(25),
int32(6),
int32(26),
int32(7),
int32(27),
int32(8),
int32(28),
int32(9),
int32(29),
int32(10),
int32(30),
int32(11),
int32(31),
int32(12),
int32(32),
int32(13),
int32(33),
int32(14),
int32(34),
int32(15),
int32(35),
int32(16),
int32(36),
int32(17),
int32(37),
int32(18),
int32(38),
int32(19),
int32(39),
int32(20),
```

```
int32(40),  
int32(22),  
int32(42),  
int32(23),  
int32(43),  
int32(24),  
int32(44),  
int32(25),  
int32(45),  
int32(26),  
int32(46),  
int32(27),  
int32(47),  
int32(28),  
int32(48),  
int32(29),  
int32(49),  
int32(30),  
int32(50),  
int32(31),  
int32(51),  
int32(32),  
int32(52),  
int32(33),  
int32(53),  
int32(34),  
int32(54),  
int32(35),  
int32(55),  
int32(36),  
int32(56),  
int32(37),  
int32(57),  
int32(38),  
int32(58),  
int32(39),  
int32(59),  
int32(40),  
int32(60),  
int32(42),  
int32(62),  
int32(43),  
int32(63),  
int32(44),  
int32(64),  
int32(45),  
int32(65),  
int32(46),  
int32(66),  
int32(47),  
int32(67),  
int32(48),  
int32(68),  
int32(49),  
int32(69),  
int32(50),  
int32(70),  
int32(51),  
int32(71),  
int32(52),  
int32(72),  
int32(53),  
int32(73),  
int32(54),  
int32(74),  
int32(55),  
int32(75),  
int32(56),  
int32(76),  
int32(57),
```

```
int32(77),  
int32(58),  
int32(78),  
int32(59),  
int32(79),  
int32(60),  
int32(80),  
int32(62),  
int32(82),  
int32(63),  
int32(83),  
int32(64),  
int32(84),  
int32(65),  
int32(85),  
int32(66),  
int32(86),  
int32(67),  
int32(87),  
int32(68),  
int32(88),  
int32(69),  
int32(89),  
int32(70),  
int32(90),  
int32(71),  
int32(91),  
int32(72),  
int32(92),  
int32(73),  
int32(93),  
int32(74),  
int32(94),  
int32(75),  
int32(95),  
int32(76),  
int32(96),  
int32(77),  
int32(97),  
int32(78),  
int32(98),  
int32(79),  
int32(99),  
int32(80),  
int32(100),  
int32(82),  
int32(102),  
int32(83),  
int32(103),  
int32(84),  
int32(104),  
int32(85),  
int32(105),  
int32(86),  
int32(106),  
int32(87),  
int32(107),  
int32(88),  
int32(108),  
int32(89),  
int32(109),  
int32(90),  
int32(110),  
int32(91),  
int32(111),  
int32(92),  
int32(112),  
int32(93),  
int32(113),  
int32(94),
```

```
int32(114),
int32(95),
int32(115),
int32(96),
int32(116),
int32(97),
int32(117),
int32(98),
int32(118),
int32(99),
int32(119),
int32(100),
int32(120),
int32(102),
int32(122),
int32(103),
int32(123),
int32(104),
int32(124),
int32(105),
int32(125),
int32(106),
int32(126),
int32(107),
int32(127),
int32(108),
int32(128),
int32(109),
int32(129),
int32(110),
int32(130),
int32(111),
int32(131),
int32(112),
int32(132),
int32(113),
int32(133),
int32(114),
int32(134),
int32(115),
int32(135),
int32(116),
int32(136),
int32(117),
int32(137),
int32(118),
int32(138),
int32(119),
int32(139),
int32(120),
int32(140),
int32(122),
int32(142),
int32(123),
int32(143),
int32(124),
int32(144),
int32(125),
int32(145),
int32(126),
int32(146),
int32(127),
int32(147),
int32(128),
int32(148),
int32(129),
int32(149),
int32(130),
int32(150),
int32(131),
```

```
int32(151),  
int32(132),  
int32(152),  
int32(133),  
int32(153),  
int32(134),  
int32(154),  
int32(135),  
int32(155),  
int32(136),  
int32(156),  
int32(137),  
int32(157),  
int32(138),  
int32(158),  
int32(139),  
int32(159),  
int32(140),  
int32(160),  
int32(142),  
int32(162),  
int32(143),  
int32(163),  
int32(144),  
int32(164),  
int32(145),  
int32(165),  
int32(146),  
int32(166),  
int32(147),  
int32(167),  
int32(148),  
int32(168),  
int32(149),  
int32(169),  
int32(150),  
int32(170),  
int32(151),  
int32(171),  
int32(152),  
int32(172),  
int32(153),  
int32(173),  
int32(154),  
int32(174),  
int32(155),  
int32(175),  
int32(156),  
int32(176),  
int32(157),  
int32(177),  
int32(158),  
int32(178),  
int32(159),  
int32(179),  
int32(160),  
int32(180),  
int32(162),  
int32(182),  
int32(163),  
int32(183),  
int32(164),  
int32(184),  
int32(165),  
int32(185),  
int32(166),  
int32(186),  
int32(167),  
int32(187),  
int32(168),
```

```
int32(188),
int32(169),
int32(189),
int32(170),
int32(190),
int32(171),
int32(191),
int32(172),
int32(192),
int32(173),
int32(193),
int32(174),
int32(194),
int32(175),
int32(195),
int32(176),
int32(196),
int32(177),
int32(197),
int32(178),
int32(198),
int32(179),
int32(199),
int32(180),
int32(200),
int32(182),
int32(202),
int32(183),
int32(203),
int32(184),
int32(204),
int32(185),
int32(205),
int32(186),
int32(206),
int32(187),
int32(207),
int32(188),
int32(208),
int32(189),
int32(209),
int32(190),
int32(210),
int32(191),
int32(211),
int32(192),
int32(212),
int32(193),
int32(213),
int32(194),
int32(214),
int32(195),
int32(215),
int32(196),
int32(216),
int32(197),
int32(217),
int32(198),
int32(218),
int32(199),
int32(219),
int32(200),
int32(220),
int32(202),
int32(222),
int32(203),
int32(223),
int32(204),
int32(224),
int32(205),
```

```
int32(225),  
int32(206),  
int32(226),  
int32(207),  
int32(227),  
int32(208),  
int32(228),  
int32(209),  
int32(229),  
int32(210),  
int32(230),  
int32(211),  
int32(231),  
int32(212),  
int32(232),  
int32(213),  
int32(233),  
int32(214),  
int32(234),  
int32(215),  
int32(235),  
int32(216),  
int32(236),  
int32(217),  
int32(237),  
int32(218),  
int32(238),  
int32(219),  
int32(239),  
int32(220),  
int32(240),  
int32(222),  
int32(242),  
int32(223),  
int32(243),  
int32(224),  
int32(244),  
int32(225),  
int32(245),  
int32(226),  
int32(246),  
int32(227),  
int32(247),  
int32(228),  
int32(248),  
int32(229),  
int32(249),  
int32(230),  
int32(250),  
int32(231),  
int32(251),  
int32(232),  
int32(252),  
int32(233),  
int32(253),  
int32(234),  
int32(254),  
int32(235),  
int32(255),  
int32(236),  
int32(256),  
int32(237),  
int32(257),  
int32(238),  
int32(258),  
int32(239),  
int32(259),  
int32(240),  
int32(260),  
int32(242),
```

```
int32(262),  
int32(243),  
int32(263),  
int32(244),  
int32(264),  
int32(245),  
int32(265),  
int32(246),  
int32(266),  
int32(247),  
int32(267),  
int32(248),  
int32(268),  
int32(249),  
int32(269),  
int32(250),  
int32(270),  
int32(251),  
int32(271),  
int32(252),  
int32(272),  
int32(253),  
int32(273),  
int32(254),  
int32(274),  
int32(255),  
int32(275),  
int32(256),  
int32(276),  
int32(257),  
int32(277),  
int32(258),  
int32(278),  
int32(259),  
int32(279),  
int32(260),  
int32(280),  
int32(262),  
int32(282),  
int32(263),  
int32(283),  
int32(264),  
int32(284),  
int32(265),  
int32(285),  
int32(266),  
int32(286),  
int32(267),  
int32(287),  
int32(268),  
int32(288),  
int32(269),  
int32(289),  
int32(270),  
int32(290),  
int32(271),  
int32(291),  
int32(272),  
int32(292),  
int32(273),  
int32(293),  
int32(274),  
int32(294),  
int32(275),  
int32(295),  
int32(276),  
int32(296),  
int32(277),  
int32(297),  
int32(278),
```

```
int32(298),  
int32(279),  
int32(299),  
int32(280),  
int32(300),  
int32(282),  
int32(302),  
int32(283),  
int32(303),  
int32(284),  
int32(304),  
int32(285),  
int32(305),  
int32(286),  
int32(306),  
int32(287),  
int32(307),  
int32(288),  
int32(308),  
int32(289),  
int32(309),  
int32(290),  
int32(310),  
int32(291),  
int32(311),  
int32(292),  
int32(312),  
int32(293),  
int32(313),  
int32(294),  
int32(314),  
int32(295),  
int32(315),  
int32(296),  
int32(316),  
int32(297),  
int32(317),  
int32(298),  
int32(318),  
int32(299),  
int32(319),  
int32(300),  
int32(320),  
int32(302),  
int32(322),  
int32(303),  
int32(323),  
int32(304),  
int32(324),  
int32(305),  
int32(325),  
int32(306),  
int32(326),  
int32(307),  
int32(327),  
int32(308),  
int32(328),  
int32(309),  
int32(329),  
int32(310),  
int32(330),  
int32(311),  
int32(331),  
int32(312),  
int32(332),  
int32(313),  
int32(333),  
int32(314),  
int32(334),  
int32(315),
```

```
int32(335),
int32(316),
int32(336),
int32(317),
int32(337),
int32(318),
int32(338),
int32(319),
int32(339),
int32(320),
int32(340),
int32(322),
int32(342),
int32(323),
int32(343),
int32(324),
int32(344),
int32(325),
int32(345),
int32(326),
int32(346),
int32(327),
int32(347),
int32(328),
int32(348),
int32(329),
int32(349),
int32(330),
int32(350),
int32(331),
int32(351),
int32(332),
int32(352),
int32(333),
int32(353),
int32(334),
int32(354),
int32(335),
int32(355),
int32(336),
int32(356),
int32(337),
int32(357),
int32(338),
int32(358),
int32(339),
int32(359),
int32(340),
int32(360),
int32(342),
int32(362),
int32(343),
int32(363),
int32(344),
int32(364),
int32(345),
int32(365),
int32(346),
int32(366),
int32(347),
int32(367),
int32(348),
int32(368),
int32(349),
int32(369),
int32(350),
int32(370),
int32(351),
int32(371),
int32(352),
```

```
int32(372),  
int32(353),  
int32(373),  
int32(354),  
int32(374),  
int32(355),  
int32(375),  
int32(356),  
int32(376),  
int32(357),  
int32(377),  
int32(358),  
int32(378),  
int32(359),  
int32(379),  
int32(360),  
int32(380),  
int32(362),  
int32(382),  
int32(363),  
int32(383),  
int32(364),  
int32(384),  
int32(365),  
int32(385),  
int32(366),  
int32(386),  
int32(367),  
int32(387),  
int32(368),  
int32(388),  
int32(369),  
int32(389),  
int32(370),  
int32(390),  
int32(371),  
int32(391),  
int32(372),  
int32(392),  
int32(373),  
int32(393),  
int32(374),  
int32(394),  
int32(375),  
int32(395),  
int32(376),  
int32(396),  
int32(377),  
int32(397),  
int32(378),  
int32(398),  
int32(379),  
int32(399),  
int32(380),  
int32(400)];  
conni(3,1:722) = [int32(22),  
int32(21),  
int32(23),  
int32(22),  
int32(24),  
int32(23),  
int32(25),  
int32(24),  
int32(26),  
int32(25),  
int32(27),  
int32(26),  
int32(28),  
int32(27),  
int32(29),
```

```
int32(28),  
int32(30),  
int32(29),  
int32(31),  
int32(30),  
int32(32),  
int32(31),  
int32(33),  
int32(32),  
int32(34),  
int32(33),  
int32(35),  
int32(34),  
int32(36),  
int32(35),  
int32(37),  
int32(36),  
int32(38),  
int32(37),  
int32(39),  
int32(38),  
int32(40),  
int32(39),  
int32(42),  
int32(41),  
int32(43),  
int32(42),  
int32(44),  
int32(43),  
int32(45),  
int32(44),  
int32(46),  
int32(45),  
int32(47),  
int32(46),  
int32(48),  
int32(47),  
int32(49),  
int32(48),  
int32(50),  
int32(49),  
int32(51),  
int32(50),  
int32(52),  
int32(51),  
int32(53),  
int32(52),  
int32(54),  
int32(53),  
int32(55),  
int32(54),  
int32(56),  
int32(55),  
int32(57),  
int32(56),  
int32(58),  
int32(57),  
int32(59),  
int32(58),  
int32(60),  
int32(59),  
int32(62),  
int32(61),  
int32(63),  
int32(62),  
int32(64),  
int32(63),  
int32(65),  
int32(64),  
int32(66),
```

```
int32(65),  
int32(67),  
int32(66),  
int32(68),  
int32(67),  
int32(69),  
int32(68),  
int32(70),  
int32(69),  
int32(71),  
int32(70),  
int32(72),  
int32(71),  
int32(73),  
int32(72),  
int32(74),  
int32(73),  
int32(75),  
int32(74),  
int32(76),  
int32(75),  
int32(77),  
int32(76),  
int32(78),  
int32(77),  
int32(79),  
int32(78),  
int32(80),  
int32(79),  
int32(82),  
int32(81),  
int32(83),  
int32(82),  
int32(84),  
int32(83),  
int32(85),  
int32(84),  
int32(86),  
int32(85),  
int32(87),  
int32(86),  
int32(88),  
int32(87),  
int32(89),  
int32(88),  
int32(90),  
int32(89),  
int32(91),  
int32(90),  
int32(92),  
int32(91),  
int32(93),  
int32(92),  
int32(94),  
int32(93),  
int32(95),  
int32(94),  
int32(96),  
int32(95),  
int32(97),  
int32(96),  
int32(98),  
int32(97),  
int32(99),  
int32(98),  
int32(100),  
int32(99),  
int32(102),  
int32(101),  
int32(103),
```

```
int32(102),  
int32(104),  
int32(103),  
int32(105),  
int32(104),  
int32(106),  
int32(105),  
int32(107),  
int32(106),  
int32(108),  
int32(107),  
int32(109),  
int32(108),  
int32(110),  
int32(109),  
int32(111),  
int32(110),  
int32(112),  
int32(111),  
int32(113),  
int32(112),  
int32(114),  
int32(113),  
int32(115),  
int32(114),  
int32(116),  
int32(115),  
int32(117),  
int32(116),  
int32(118),  
int32(117),  
int32(119),  
int32(118),  
int32(120),  
int32(119),  
int32(122),  
int32(121),  
int32(123),  
int32(122),  
int32(124),  
int32(123),  
int32(125),  
int32(124),  
int32(126),  
int32(125),  
int32(127),  
int32(126),  
int32(128),  
int32(127),  
int32(129),  
int32(128),  
int32(130),  
int32(129),  
int32(131),  
int32(130),  
int32(132),  
int32(131),  
int32(133),  
int32(132),  
int32(134),  
int32(133),  
int32(135),  
int32(134),  
int32(136),  
int32(135),  
int32(137),  
int32(136),  
int32(138),  
int32(137),  
int32(139),
```

```
int32(138),  
int32(140),  
int32(139),  
int32(142),  
int32(141),  
int32(143),  
int32(142),  
int32(144),  
int32(143),  
int32(145),  
int32(144),  
int32(146),  
int32(145),  
int32(147),  
int32(146),  
int32(148),  
int32(147),  
int32(149),  
int32(148),  
int32(150),  
int32(149),  
int32(151),  
int32(150),  
int32(152),  
int32(151),  
int32(153),  
int32(152),  
int32(154),  
int32(153),  
int32(155),  
int32(154),  
int32(156),  
int32(155),  
int32(157),  
int32(156),  
int32(158),  
int32(157),  
int32(159),  
int32(158),  
int32(160),  
int32(159),  
int32(162),  
int32(161),  
int32(163),  
int32(162),  
int32(164),  
int32(163),  
int32(165),  
int32(164),  
int32(166),  
int32(165),  
int32(167),  
int32(166),  
int32(168),  
int32(167),  
int32(169),  
int32(168),  
int32(170),  
int32(169),  
int32(171),  
int32(170),  
int32(172),  
int32(171),  
int32(173),  
int32(172),  
int32(174),  
int32(173),  
int32(175),  
int32(174),  
int32(176),
```

```
int32(175),
int32(177),
int32(176),
int32(178),
int32(177),
int32(179),
int32(178),
int32(180),
int32(179),
int32(182),
int32(181),
int32(183),
int32(182),
int32(184),
int32(183),
int32(185),
int32(184),
int32(186),
int32(185),
int32(187),
int32(186),
int32(188),
int32(187),
int32(189),
int32(188),
int32(190),
int32(189),
int32(191),
int32(190),
int32(192),
int32(191),
int32(193),
int32(192),
int32(194),
int32(193),
int32(195),
int32(194),
int32(196),
int32(195),
int32(197),
int32(196),
int32(198),
int32(197),
int32(199),
int32(198),
int32(200),
int32(199),
int32(202),
int32(201),
int32(203),
int32(202),
int32(204),
int32(203),
int32(205),
int32(204),
int32(206),
int32(205),
int32(207),
int32(206),
int32(208),
int32(207),
int32(209),
int32(208),
int32(210),
int32(209),
int32(211),
int32(210),
int32(212),
int32(211),
int32(213),
```

```
int32(212),  
int32(214),  
int32(213),  
int32(215),  
int32(214),  
int32(216),  
int32(215),  
int32(217),  
int32(216),  
int32(218),  
int32(217),  
int32(219),  
int32(218),  
int32(220),  
int32(219),  
int32(222),  
int32(221),  
int32(223),  
int32(222),  
int32(224),  
int32(223),  
int32(225),  
int32(224),  
int32(226),  
int32(225),  
int32(227),  
int32(226),  
int32(228),  
int32(227),  
int32(229),  
int32(228),  
int32(230),  
int32(229),  
int32(231),  
int32(230),  
int32(232),  
int32(231),  
int32(233),  
int32(232),  
int32(234),  
int32(233),  
int32(235),  
int32(234),  
int32(236),  
int32(235),  
int32(237),  
int32(236),  
int32(238),  
int32(237),  
int32(239),  
int32(238),  
int32(240),  
int32(239),  
int32(242),  
int32(241),  
int32(243),  
int32(242),  
int32(244),  
int32(243),  
int32(245),  
int32(244),  
int32(246),  
int32(245),  
int32(247),  
int32(246),  
int32(248),  
int32(247),  
int32(249),  
int32(248),  
int32(250),
```

```
int32(249),
int32(251),
int32(250),
int32(252),
int32(251),
int32(253),
int32(252),
int32(254),
int32(253),
int32(255),
int32(254),
int32(256),
int32(255),
int32(257),
int32(256),
int32(258),
int32(257),
int32(259),
int32(258),
int32(260),
int32(259),
int32(262),
int32(261),
int32(263),
int32(262),
int32(264),
int32(263),
int32(265),
int32(264),
int32(266),
int32(265),
int32(267),
int32(266),
int32(268),
int32(267),
int32(269),
int32(268),
int32(270),
int32(269),
int32(271),
int32(270),
int32(272),
int32(271),
int32(273),
int32(272),
int32(274),
int32(273),
int32(275),
int32(274),
int32(276),
int32(275),
int32(277),
int32(276),
int32(278),
int32(277),
int32(279),
int32(278),
int32(280),
int32(279),
int32(282),
int32(281),
int32(283),
int32(282),
int32(284),
int32(283),
int32(285),
int32(284),
int32(286),
int32(285),
int32(287),
```

```
int32(286),  
int32(288),  
int32(287),  
int32(289),  
int32(288),  
int32(290),  
int32(289),  
int32(291),  
int32(290),  
int32(292),  
int32(291),  
int32(293),  
int32(292),  
int32(294),  
int32(293),  
int32(295),  
int32(294),  
int32(296),  
int32(295),  
int32(297),  
int32(296),  
int32(298),  
int32(297),  
int32(299),  
int32(298),  
int32(300),  
int32(299),  
int32(302),  
int32(301),  
int32(303),  
int32(302),  
int32(304),  
int32(303),  
int32(305),  
int32(304),  
int32(306),  
int32(305),  
int32(307),  
int32(306),  
int32(308),  
int32(307),  
int32(309),  
int32(308),  
int32(310),  
int32(309),  
int32(311),  
int32(310),  
int32(312),  
int32(311),  
int32(313),  
int32(312),  
int32(314),  
int32(313),  
int32(315),  
int32(314),  
int32(316),  
int32(315),  
int32(317),  
int32(316),  
int32(318),  
int32(317),  
int32(319),  
int32(318),  
int32(320),  
int32(319),  
int32(322),  
int32(321),  
int32(323),  
int32(322),  
int32(324),
```

```
int32(323),
int32(325),
int32(324),
int32(326),
int32(325),
int32(327),
int32(326),
int32(328),
int32(327),
int32(329),
int32(328),
int32(330),
int32(329),
int32(331),
int32(330),
int32(332),
int32(331),
int32(333),
int32(332),
int32(334),
int32(333),
int32(335),
int32(334),
int32(336),
int32(335),
int32(337),
int32(336),
int32(338),
int32(337),
int32(339),
int32(338),
int32(340),
int32(339),
int32(342),
int32(341),
int32(343),
int32(342),
int32(344),
int32(343),
int32(345),
int32(344),
int32(346),
int32(345),
int32(347),
int32(346),
int32(348),
int32(347),
int32(349),
int32(348),
int32(350),
int32(349),
int32(351),
int32(350),
int32(352),
int32(351),
int32(353),
int32(352),
int32(354),
int32(353),
int32(355),
int32(354),
int32(356),
int32(355),
int32(357),
int32(356),
int32(358),
int32(357),
int32(359),
int32(358),
int32(360),
```

```
int32(359),  
int32(362),  
int32(361),  
int32(363),  
int32(362),  
int32(364),  
int32(363),  
int32(365),  
int32(364),  
int32(366),  
int32(365),  
int32(367),  
int32(366),  
int32(368),  
int32(367),  
int32(369),  
int32(368),  
int32(370),  
int32(369),  
int32(371),  
int32(370),  
int32(372),  
int32(371),  
int32(373),  
int32(372),  
int32(374),  
int32(373),  
int32(375),  
int32(374),  
int32(376),  
int32(375),  
int32(377),  
int32(376),  
int32(378),  
int32(377),  
int32(379),  
int32(378),  
int32(380),  
int32(379),  
int32(382),  
int32(381),  
int32(383),  
int32(382),  
int32(384),  
int32(383),  
int32(385),  
int32(384),  
int32(386),  
int32(385),  
int32(387),  
int32(386),  
int32(388),  
int32(387),  
int32(389),  
int32(388),  
int32(390),  
int32(389),  
int32(391),  
int32(390),  
int32(392),  
int32(391),  
int32(393),  
int32(392),  
int32(394),  
int32(393),  
int32(395),  
int32(394),  
int32(396),  
int32(395),  
int32(397),
```

```
int32(396),  
int32(398),  
int32(397),  
int32(399),  
int32(398),  
int32(400),  
int32(399)];  
itrace = int32(0);  
[cooriOut, edgeiOut, conniOut, cooro, edgeo, conno, ifail] = ...  
    d06da(itype, trans, coori, edgei, conni, itrace)
```

```
cooriOut =  
    array elided  
edgeiOut =  
    array elided  
conniOut =  
    array elided  
cooro =  
    array elided  
edgeo =  
    array elided  
conno =  
    array elided  
ifail =  
    0
```
