Exercise 7.1 C-code for a R_{ab} for a generic metric

```
{x,y,z}::Coordinate.
{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u#}::Indices(values={x,y,z},position=independent).

\partial{#}::PartialDerivative;

g_{a b}::Metric;
g^{a b}::InverseMetric;

import cdblib

FourRab = cdblib.get ('FourRab','ex-0309.json')

Rab := 1/4 @(FourRab).

substitute (Rab, $ \partial_{a b}{g_{c d}} -> dg_{c d a b} $)
substitute (Rab, $ \partial_{a}{g_{b c}}   -> dg_{b c a}   $)

# ------------------------------------------------------------------
# build rules to export Cadabra expressions to Python
# use known symmetries for g_{a b}, dg_{ab,c,d} etc.
# note: replacements must not contain underscores (reserved for subscripts),
#       so g_{x x}-> g_xx is not allowed

gabRule := {g_{x x} -> gxx, g_{x y} -> gxy, g_{x z} -> gxz,
            g_{y x} -> gxy, g_{y y} -> gyy, g_{y z} -> gyz,
            g_{z x} -> gxz, g_{z y} -> gyz, g_{z z} -> gzz}.

iabRule := {g^{x x} -> ixx, g^{x y} -> ixy, g^{x z} -> ixz,
            g^{y x} -> ixy, g^{y y} -> iyy, g^{y z} -> iyz,
            g^{z x} -> ixz, g^{z y} -> iyz, g^{z z} -> izz}.

d1gabRule := {dg_{x x x} -> dgxxx, dg_{x y x} -> dgxyx, dg_{x z x} -> dgxzx,
              dg_{y x x} -> dgxyx, dg_{y y x} -> dgyyx, dg_{y z x} -> dgyzx,
              dg_{z x x} -> dgxzx, dg_{z y x} -> dgyzx, dg_{z z x} -> dgzzx,

              dg_{x x y} -> dgxxy, dg_{x y y} -> dgxyy, dg_{x z y} -> dgxzy,
```

```
                dg_{y x y} -> dgxyy, dg_{y y y} -> dgyyy, dg_{y z y} -> dgyzy,
                dg_{z x y} -> dgxzy, dg_{z y y} -> dgyzy, dg_{z z y} -> dgzzy,

                dg_{x x z} -> dgxxz, dg_{x y z} -> dgxyz, dg_{x z z} -> dgxzz,
                dg_{y x z} -> dgxyz, dg_{y y z} -> dgyyz, dg_{y z z} -> dgyzz,
                dg_{z x z} -> dgxzz, dg_{z y z} -> dgyzz, dg_{z z z} -> dgzzz}.

d2gabRule := {dg_{x x x x} -> dgxxxx, dg_{x y x x} -> dgxyxx, dg_{x z x x} -> dgxzxx,
                dg_{y x x x} -> dgxyxx, dg_{y y x x} -> dgyyxx, dg_{y z x x} -> dgyzxx,
                dg_{z x x x} -> dgxzxx, dg_{z y x x} -> dgyzxx, dg_{z z x x} -> dgzzxx,
                dg_{x x y x} -> dgxxyx, dg_{x y y x} -> dgxyyx, dg_{x z y x} -> dgxzyx,
                dg_{y x y x} -> dgxyyx, dg_{y y y x} -> dgyyyx, dg_{y z y x} -> dgyzyx,
                dg_{z x y x} -> dgxzyx, dg_{z y y x} -> dgyzyx, dg_{z z y x} -> dgzzyx,
                dg_{x x z x} -> dgxxzx, dg_{x y z x} -> dgxyzx, dg_{x z z x} -> dgxzzx,
                dg_{y x z x} -> dgxyzx, dg_{y y z x} -> dgyyzx, dg_{y z z x} -> dgyzzx,
                dg_{z x z x} -> dgxzzx, dg_{z y z x} -> dgyzzx, dg_{z z z x} -> dgzzzx,

                dg_{x x x y} -> dgxxxy, dg_{x y x y} -> dgxyxy, dg_{x z x y} -> dgxzxy,
                dg_{y x x y} -> dgxyxy, dg_{y y x y} -> dgyyxy, dg_{y z x y} -> dgyzxy,
                dg_{z x x y} -> dgxzxy, dg_{z y x y} -> dgyzxy, dg_{z z x y} -> dgzzxy,
                dg_{x x y y} -> dgxxyy, dg_{x y y y} -> dgxyyy, dg_{x z y y} -> dgxzyy,
                dg_{y x y y} -> dgxyyy, dg_{y y y y} -> dgyyyy, dg_{y z y y} -> dgyzyy,
                dg_{z x y y} -> dgxzyy, dg_{z y y y} -> dgyzyy, dg_{z z y y} -> dgzzyy,
                dg_{x x z y} -> dgxxzy, dg_{x y z y} -> dgxyzy, dg_{x z z y} -> dgxzzy,
                dg_{y x z y} -> dgxyzy, dg_{y y z y} -> dgyyzy, dg_{y z z y} -> dgyzzy,
                dg_{z x z y} -> dgxzzy, dg_{z y z y} -> dgyzzy, dg_{z z z y} -> dgzzzy,

                dg_{x x x z} -> dgxxxz, dg_{x y x z} -> dgxyxz, dg_{x z x z} -> dgxzxz,
                dg_{y x x z} -> dgxyxz, dg_{y y x z} -> dgyyxz, dg_{y z x z} -> dgyzxz,
                dg_{z x x z} -> dgxzxz, dg_{z y x z} -> dgyzxz, dg_{z z x z} -> dgzzxz,
                dg_{x x y z} -> dgxxyz, dg_{x y y z} -> dgxyyz, dg_{x z y z} -> dgxzyz,
                dg_{y x y z} -> dgxyyz, dg_{y y y z} -> dgyyyz, dg_{y z y z} -> dgyzyz,
                dg_{z x y z} -> dgxzyz, dg_{z y y z} -> dgyzyz, dg_{z z y z} -> dgzzyz,
                dg_{x x z z} -> dgxxzz, dg_{x y z z} -> dgxyzz, dg_{x z z z} -> dgxzzz,
                dg_{y x z z} -> dgxyzz, dg_{y y z z} -> dgyyzz, dg_{y z z z} -> dgyzzz,
                dg_{z x z z} -> dgxzzz, dg_{z y z z} -> dgyzzz, dg_{z z z z} -> dgzzzz}.

def write_code (obj,name,filename,rank):
```

```
    import os

    from sympy.printing.ccode import C99CodePrinter as printer
    from sympy.printing.codeprinter import Assignment

    idx=[]  # indices in the form [{x, x}, {x, y} ...]
    lst=[]  # corresponding terms [termxx, termxy, ...]

    for i in range( len(obj[rank]) ):                 # rank = number of free indices
        idx.append( str(obj[rank][i][0]._sympy_()) )  # indices for this term
        lst.append( str(obj[rank][i][1]._sympy_()) )  # the matching term

    mat = sympy.Matrix([lst])                         # row vector of terms
    sub_exprs, simplified_rhs = sympy.cse(mat)        # optimise code

    with open(os.getcwd() + '/' + filename, 'w') as out:

       for lhs, rhs in sub_exprs:
          out.write(printer().doprint(Assignment(lhs, rhs))+'\n')

       for index, rhs in enumerate (simplified_rhs[0]):
          lhs = sympy.Symbol(name+' '+(idx[index]).replace(', ',',]['))
          out.write(printer().doprint(Assignment(lhs, rhs))+'\n')

evaluate   (Rab,  gabRule+d1gabRule+d2gabRule+iabRule, simplify=False)

write_code (Rab, 'Rab', 'ex-0701-rab.c',2)
```

The code for R_{ab} can be found in the file `ex-0701-rab.c`. It is long and it would require more work to turn it into something useful in a numerical code. For example, functions would be needed to compute the first and second partial derivatives of the metric. But that is not a Cadabra issue.