Exercise 4.3

task a)

```
ClearAll["Global`*"];
σ = 10; r = 28; b = 8 / 3;
xdot[x_, y_, z_] := σ (y - x)
ydot[x_, y_, z_] := r x - y - x z
zdot[x_, y_, z_] := x y - b z

initCond = {x[0] == 0.1, y[0] == 0.1, z[0] == 0.1};

tmax = 100;
dt = 0.001;
numSteps = Floor[tmax / dt];
Q = IdentityMatrix[3];
R = IdentityMatrix[3];

initialConditions = {x[0] == 0.1, y[0] == 0.1, z[0] == 0.1};
timeRange = {t, 0, numSteps};
lorenzEquations = {
   x'[t] == xdot[x[t], y[t], z[t]],
   y'[t] == ydot[x[t], y[t], z[t]],
   z'[t] == zdot[x[t], y[t], z[t]],
   initialConditions};
sol = NDSolve[lorenzEquations, {x, y, z}, timeRange][[1]];

solx =  x /. sol[[1]];
soly = y /. sol[[2]];
solz = z /. sol[[3]];

J = D[{xdot[x, y, z], ydot[x, y, z], zdot[x, y, z]}, {{x, y, z}}];

Jt[t_] := J /. {x → solx[t], y → soly[t], z → solz[t]};

λAccumulate = ConstantArray[0., 3];

λ1 = ConstantArray[0, numSteps];
λ2 = ConstantArray[0, numSteps];
λ3 = ConstantArray[0, numSteps];

For[k = 1, k ≤ numSteps, k++, t = k * dt;
 M = IdentityMatrix[3] + Jt[t] dt;
 {Qnew, R} = QRDecomposition[M.Transpose[Q]];
 Q = Qnew;
 λAccumulate += Log[Abs[Diagonal[R]]];
```

```
 λ1[[k]] = λAccumulate[[1]] / t;
 λ2[[k]] = λAccumulate[[2]] / t;
 λ3[[k]] = λAccumulate[[3]] / t;
]

λ = λAccumulate / (tmax)

λ1;
λ2;
λ3;
```

Out[∘]=

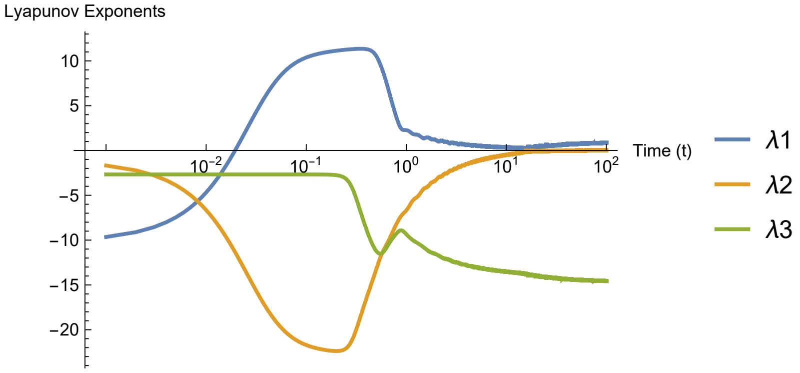
```
{0.846139, 0.0134154, -14.5593}
```

task b)

In[∘]:=

```
ListLinePlot[{
   Table[{k*dt, λ1[[k]]}, {k, 1, numSteps}],
   Table[{k*dt, λ2[[k]]}, {k, 1, numSteps}],
   Table[{k*dt, λ3[[k]]}, {k, 1, numSteps}]},
  PlotLegends → {"λ1", "λ2", "λ3"},
  AxesLabel → {"Time (t)", "Lyapunov Exponents"},
  ScalingFunctions → {"Log10", None},
  Ticks → {Table[{10^i, Superscript[10, i]},
     {i, Floor[Log10[dt]], Ceiling[Log10[tmax]]}], Automatic},
  PlotRange → All]
```

Out[∘]=



Sum of exponents should be $-\sigma$-1-b =-10-1-2.67 = -13.67, i get 0.84+0.013-14.56 =-13.71. Thus indicating convergence. (Within 5%)

task c)

```
ClearAll["Global`*"];
σ = 10; r = 28; b = 19 / 6;
xdot[x_, y_, z_] := σ (y - x)
ydot[x_, y_, z_] := r x - y - x z
```

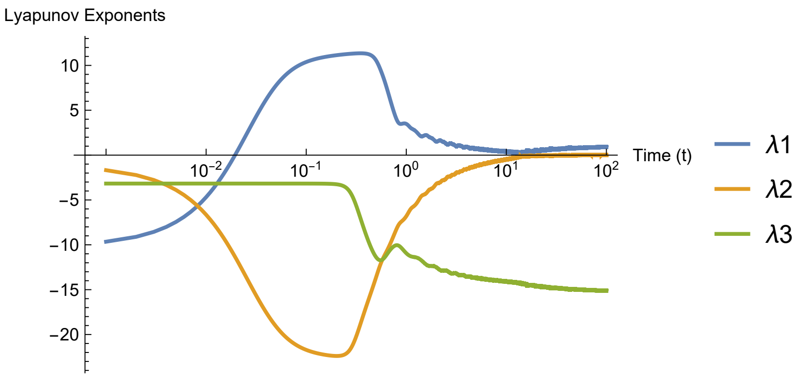
```
zdot[x_, y_, z_] := x y - b z

tmax = 100;
dt = 0.001;
numSteps = Floor[tmax / dt];
Q = IdentityMatrix[3];
R = IdentityMatrix[3];

initialConditions = {x[0] == 0.1, y[0] == 0.1, z[0] == 0.1};
timeRange = {t, 0, numSteps};
lorenzEquations = {
   x'[t] == xdot[x[t], y[t], z[t]],
   y'[t] == ydot[x[t], y[t], z[t]],
   z'[t] == zdot[x[t], y[t], z[t]],
   initialConditions};
sol = NDSolve[lorenzEquations, {x, y, z}, timeRange][[1]];

solx =  x /. sol[[1]];
soly = y /. sol[[2]];
solz = z /. sol[[3]];

J = D[{xdot[x, y, z], ydot[x, y, z], zdot[x, y, z]}, {{x, y, z}}];

Jt[t_] := J /. {x → solx[t], y → soly[t], z → solz[t]};

λAccumulate = ConstantArray[0., 3];

λ1 = ConstantArray[0, numSteps];
λ2 = ConstantArray[0, numSteps];
λ3 = ConstantArray[0, numSteps];

For[k = 1, k ≤ numSteps, k++, t = k * dt;
 M = IdentityMatrix[3] + Jt[t] dt;
 {Qnew, R} = QRDecomposition[M.Transpose[Q]];
 Q = Qnew;
 λAccumulate += Log[Abs[Diagonal[R]]];

 λ1[[k]] = λAccumulate[[1]] / t;
 λ2[[k]] = λAccumulate[[2]] / t;
 λ3[[k]] = λAccumulate[[3]] / t;
]

λ = λAccumulate / (tmax)

λ1;
λ2;
λ3;

ListLinePlot[{
  Table[{k * dt, λ1[[k]]}, {k, 1, numSteps}],
```

```
  Table[{k*dt, λ2[[k]]}, {k, 1, numSteps}],
  Table[{k*dt, λ3[[k]]}, {k, 1, numSteps}]},
 PlotLegends → {"λ1", "λ2", "λ3"},
 AxesLabel → {"Time (t)", "Lyapunov Exponents"},
 ScalingFunctions → {"Log10", None},
 Ticks → {Table[{10^i, Superscript[10, i]},
    {i, Floor[Log10[dt]], Ceiling[Log10[tmax]]}], Automatic},
 PlotRange → All]
```

Out[∘]=

```
{0.893786, 0.0277694, -15.1231}
```

Out[∘]=



sum of exponents should be: -10-1-3.17 = -14.17, i get: 0.89+0.028-15.12 = -14.2 which also indicate convergence. (Within 5%)

task d)

In[∘]:=

```
λ = λAccumulate / (tmax)
```

Out[∘]=

```
{0.893786, 0.0277694, -15.1231}
```

task e)

```
ClearAll["Global`*"];
σ = 16;
r = 330;
b = 5;
xdot[x_, y_, z_] := σ * (y - x)
ydot[x_, y_, z_] := r * x - y - x * z
zdot[x_, y_, z_] := x * y - b * z


tmax = 15 000;
dt = 0.0006;
numSteps = Floor[tmax / dt];
Q = IdentityMatrix[3];
R = IdentityMatrix[3];

sol = NDSolve[{
```

```
      x'[t] == xdot[x[t], y[t], z[t]],
      y'[t] == ydot[x[t], y[t], z[t]],
      z'[t] == zdot[x[t], y[t], z[t]],
      x[0] == 0.01,
      y[0] == 0.01,
      z[0] == 0.01},
     {x, y, z}, {t, 0, tmax}, Method → "StiffnessSwitching"][[1]];

Print["NDSolve complete"]
solx =  x /. sol[[1]];
soly = y /. sol[[2]];
solz = z /. sol[[3]];

J = D[{xdot[x, y, z], ydot[x, y, z], zdot[x, y, z]}, {{x, y, z}}];
Jt[t_] := J /. {x → solx[t], y → soly[t], z → solz[t]};

λAccumulate = ConstantArray[0., 3];

λ1 = ConstantArray[0., numSteps];
λ2 = ConstantArray[0., numSteps];
λ3 = ConstantArray[0., numSteps];
id = IdentityMatrix[3];

For[k = 1, k ≤ numSteps, k++, t = k * dt;
  M = id + Jt[t] * dt;
  Z = M.Q;
  {Qnew, R} = QRDecomposition[Z];
  Q = Transpose[Qnew];
  λAccumulate += Log[Abs[Diagonal[R]]];

  λ1[[k]] = λAccumulate[[1]] / (t + dt);
  λ2[[k]] = λAccumulate[[2]] / (t + dt);
  λ3[[k]] = λAccumulate[[3]] / (t + dt);
 ];

λ = λAccumulate / tmax

λ1;
λ2;
λ3;

Print["Plot next"]
ListLinePlot[{
  Table[{k * dt, λ1[[k]]}, {k, 1, numSteps, 100}],
  Table[{k * dt, λ2[[k]]}, {k, 1, numSteps, 100}],
  Table[{k * dt, λ3[[k]]}, {k, 1, numSteps, 100}]},
 PlotLegends → {"λ1", "λ2", "λ3"},
 AxesLabel → {"Time (t)", "Lyapunov Exponents"},
 ScalingFunctions → {"Log10", None}, Ticks →
  {Table[{10^i, Superscript[10, i]}, {i, Floor[Log10[dt]], Ceiling[Log10[tmax]]}],
   Automatic}, PlotRange → All]
```

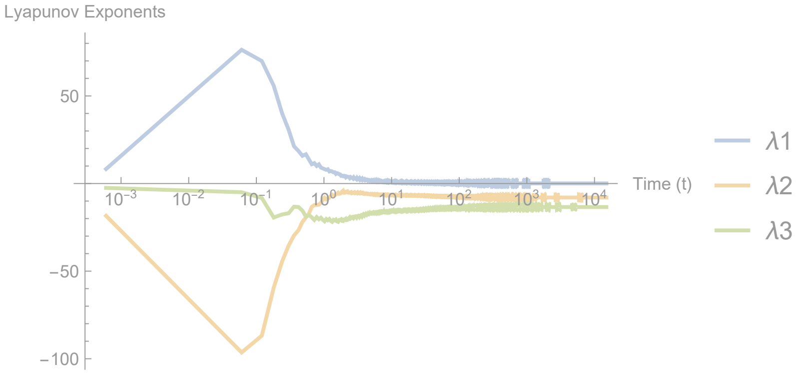
```
NDSolve complete
```

Out[1834]=

```
{0.0671622, -7.97972, -13.413}
```

```
Plot next
```

Out[1839]=



Checking the sum of Lyapunov exponents we get −21.3255578 which is indeed within 5% precision of -22, thus i believe the exponents have converged.

task f)

In[1840]:=

```
λ = λAccumulate / tmax
```

Out[1840]=

```
{0.0671622, -7.97972, -13.413}
```

Again looking at the sum, I believe these exponents should be correct, but OpenTA disagree. Furthermore, since the first one is positive, this indicates chaotic behaviour.