Harper — PS 10 — 2025-02-25

EIWL Sections 26-28

Looks very good. 10 / 10.

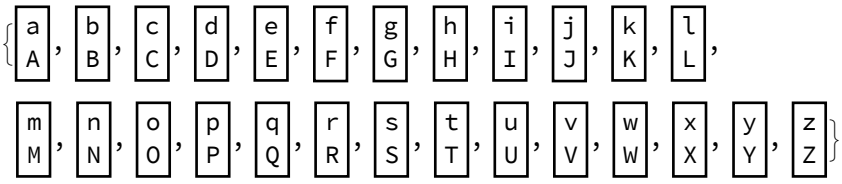
A few comments (some in response to your comments) below.

Section 26

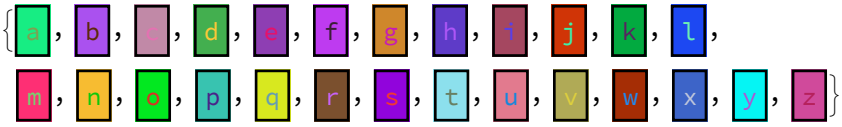
In[243]:=
```
Power[#, 2] & /@ Range[20]
```
Out[243]=
```
{1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400}
```

In[244]:=
```
Blend[{#, Red}] & /@ {Yellow, Green, Blue}
```
Out[244]=

In[245]:=
```
Framed[Column[{#, ToUpperCase[#]}]] & /@ FromLetterNumber[Range[26]]
```
Out[245]=



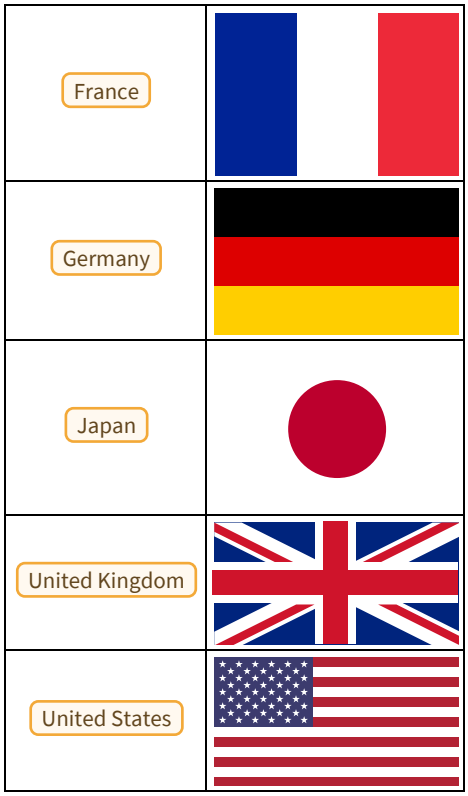
In[246]:=
```
Framed[Style[#, RandomColor[]], Background → RandomColor[]] & /@
 FromLetterNumber[Range[26]]
```
Out[246]=



In[247]:=

```
Grid[{#, EntityValue[#, "Flag"]} & /@ EntityList[Group of 5 COUNTRIES], Frame → All]
```

Out[247]=



In[248]:=

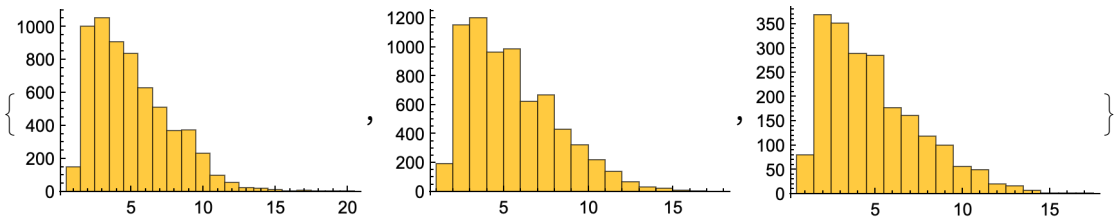
```
WordCloud[WikipediaData[#]] & /@ {"apple", "peach", "pear"}
```

Out[248]=

In[249]:=

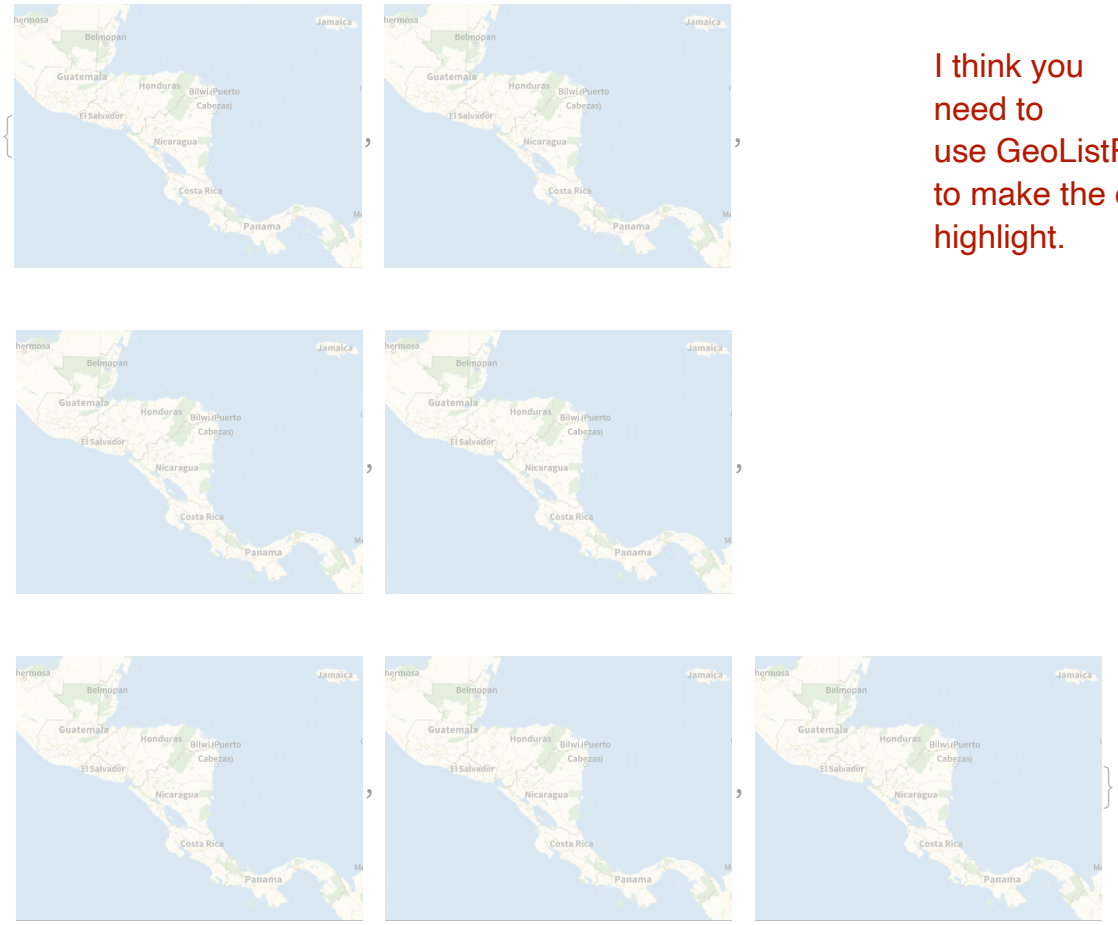
```
Histogram[StringLength[TextWords[WikipediaData[#]]]] & /@ {"apple", "peach", "pear"}
```

Out[249]=



```
GeoGraphics[{#}, GeoRange → EntityList["Central America" COUNTRIES]] & /@
  EntityList["Central America" COUNTRIES]
```

Out[250]=



I think you need to use GeoListPlot to make the countries highlight.

In[251]:=

```
Range[10]^2 + 1
```

Out[251]=

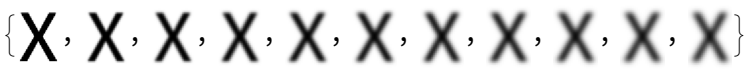
```
{2, 5, 10, 17, 26, 37, 50, 65, 82, 101}
```

Section 27

In[252]:=

```
NestList[Blur, Rasterize[Style["X", 30]], 10]
```

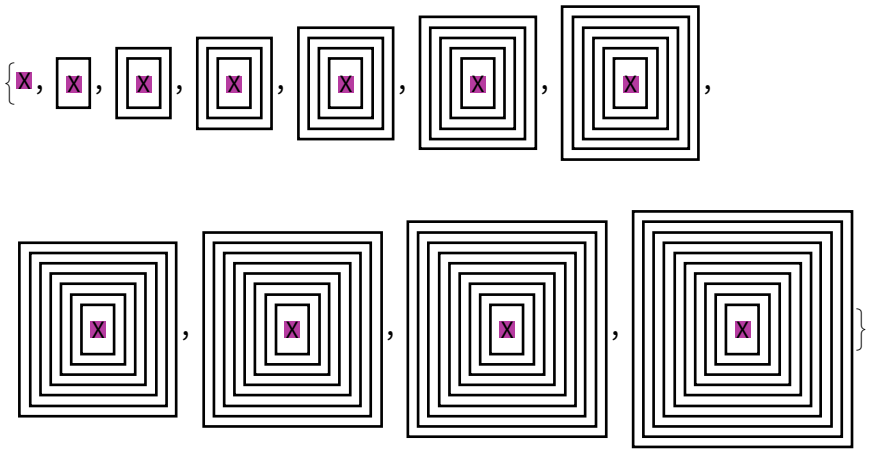
Out[252]=



In[253]:=

```
NestList[Framed, Style["X", Background → RandomColor[]], 10]
```

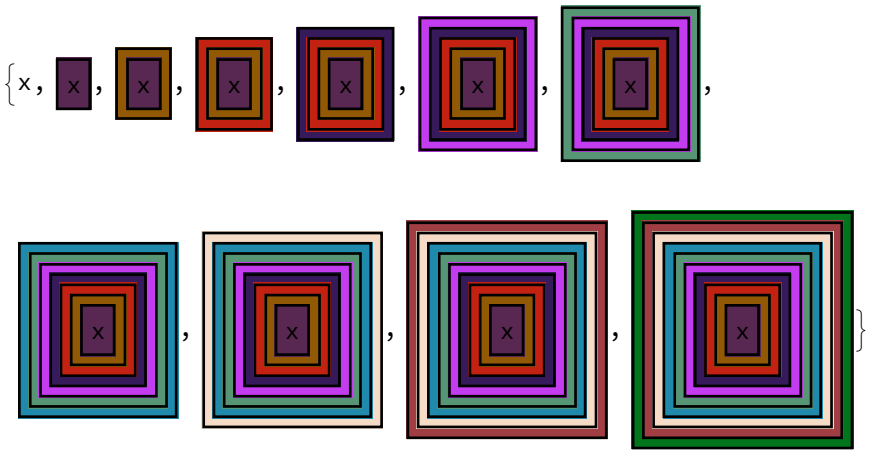
Out[253]=



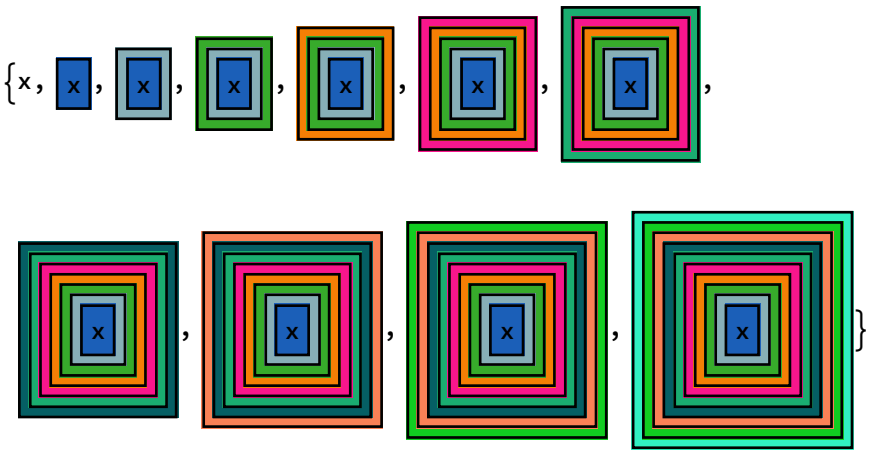
In[254]:=

```
NestList[Framed[#, Background → RandomColor[]] &, x, 10]
```

Out[254]=

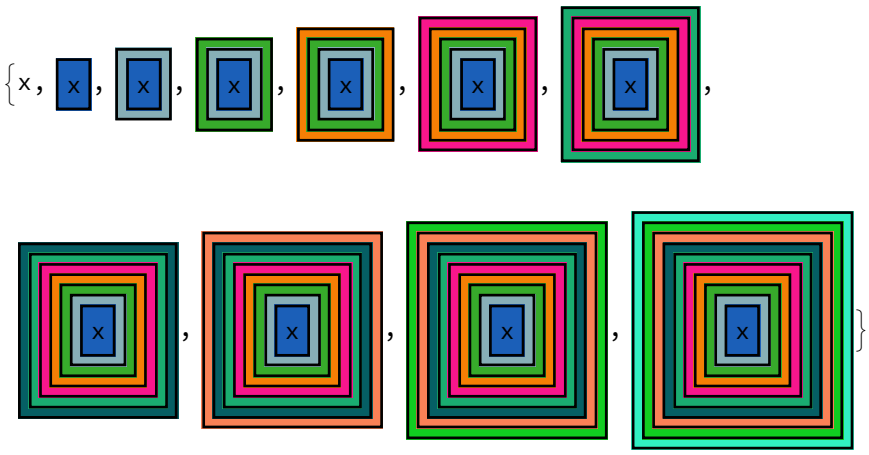


In[255]:=



```
(*I don't understand how the above function avoids the problem of the previous*)
```

Out[255]=

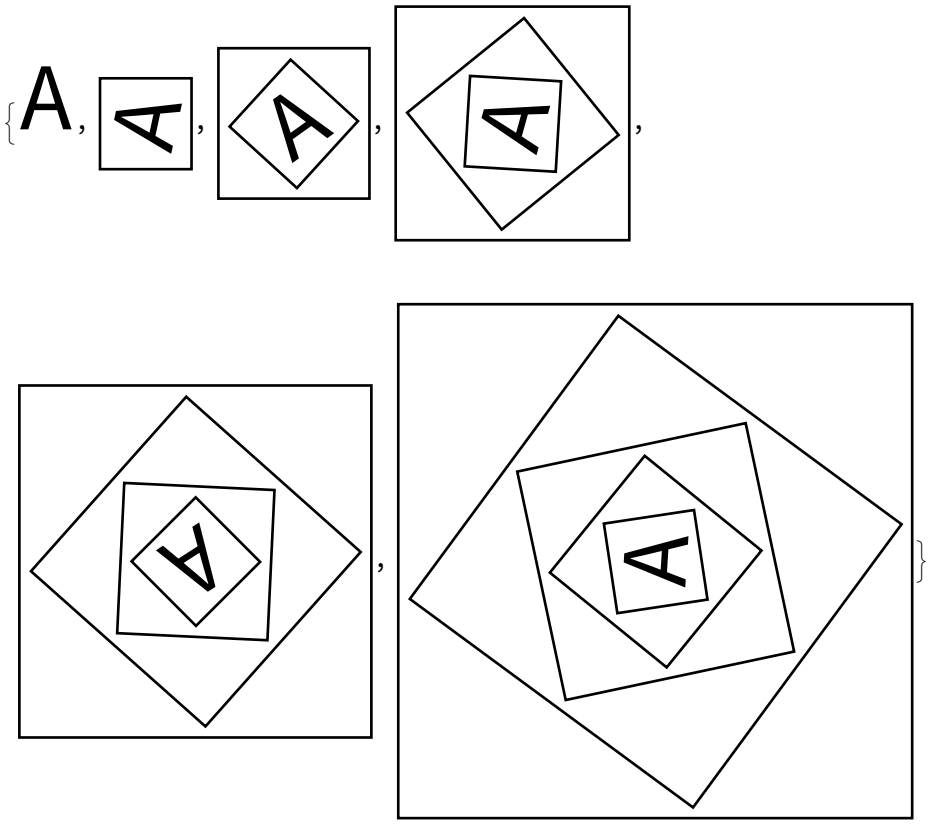


Is the difference that in one case,
NestList is repeatedly applying
Framed whereas in the other case,
it is applying Framed with a new background color?
That seems pretty understandable, but maybe
I don't understand your question.

In[256]:=

```
NestList[Framed[Rotate[#, RandomReal[{0, 360 °}]]] &, Style["A", 50], 5]
```

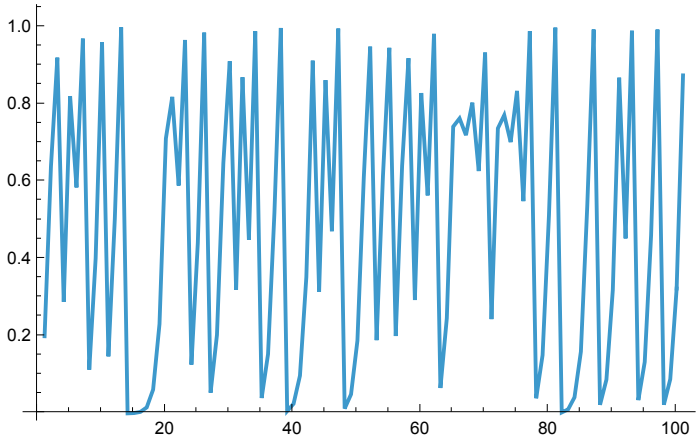
Out[256]=



In[257]:=

```
ListLinePlot[NestList[4 # (1 - #) &, 0.2, 100]]
```

Out[257]=



In[258]:=

```
N[Nest[1 + 1 / # &, 1, 30]]
```

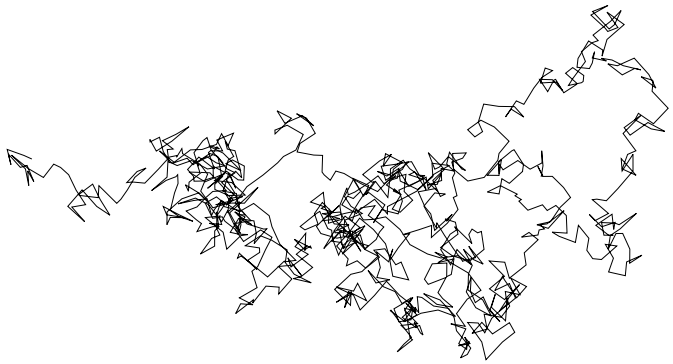
Out[258]=

```
1.61803
```

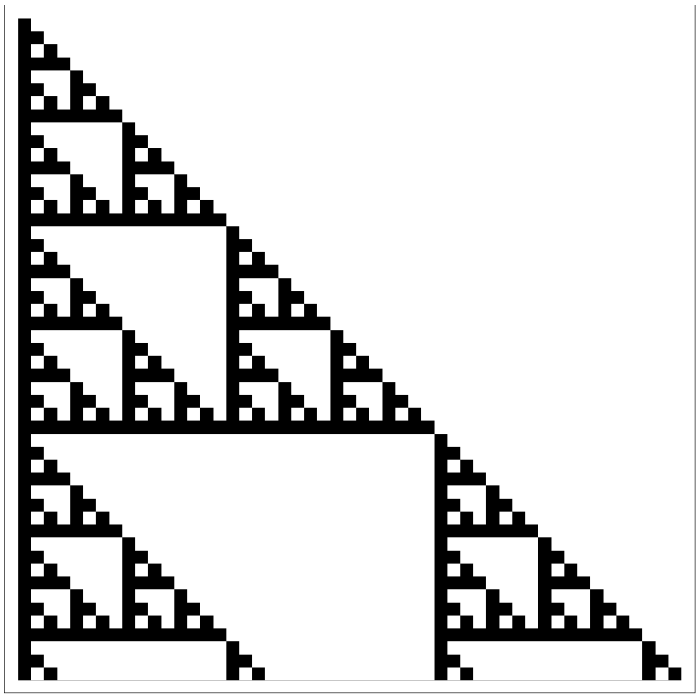
In[259]:=
```
NestList[3 * # &, 1, 10]
```
Out[259]=
$$\{1, 3, 9, 27, 81, 243, 729, 2187, 6561, 19\,683, 59\,049\}$$

In[260]:=
```
NestList[(# + 2 / #) / 2 &, 1.0, 5] - Sqrt[2]
```
Out[260]=
$$\left\{-0.414214, 0.0857864, 0.0024531, 2.1239 \times 10^{-6}, 1.59472 \times 10^{-12}, -2.22045 \times 10^{-16}\right\}$$

In[261]:=
```
Graphics[Line[NestList[# + RandomReal[{-1, 1}, 2] &, {0, 0}, 1000]]]
```
Out[261]=

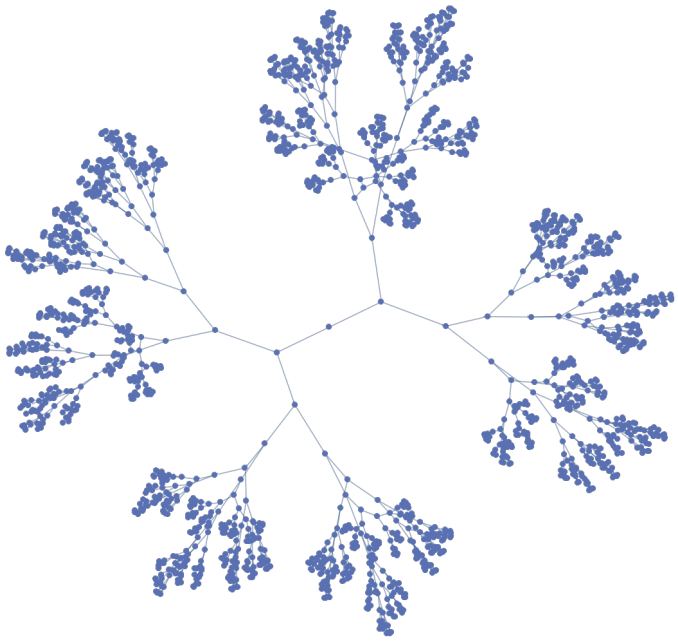


In[262]:=
```
ArrayPlot[NestList[Mod[Join[{0}, #] + Join[#, {0}], 2] &, {1}, 50]]
```
Out[262]=



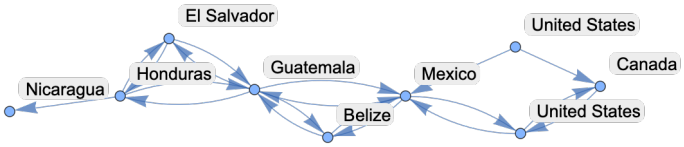
In[263]:=
```
NestGraph[{# → # + 1, # → 2 #} &, 0, 10]
```

Out[263]=



In[264]:=
```
NestGraph[#["BorderingCountries"] &, Entity["Country", "USA"], 4, VertexLabels → All]
```

Out[264]=



Section 28

In[265]:=
```
123^321 > 456^123
```

Out[265]=
```
True
```

In[266]:=
```
Select[Range[100], Total[IntegerDigits[#]] < 5 &]
```

Out[266]=
```
{1, 2, 3, 4, 10, 11, 12, 13, 20, 21, 22, 30, 31, 40, 100}
```

In[267]:=
```
If[PrimeQ[#], Style[#, Red], #] & /@ Range[20]
```

Out[267]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

In[268]:=
```
Select[WordList[], StringTake[#, 1] == StringTake[StringReverse[#], 1] == "p" &]
```
Out[268]=
```
{pap, paperclip, parsnip, partisanship, partnership, pawnshop, peep, penmanship,
 pep, pickup, pileup, pip, plop, plump, polyp, pomp, pop, premiership,
 prep, primp, professorship, prop, proprietorship, pulp, pump, pup}
```

In[269]:=
```
Select[Prime[Range[100]], Last[IntegerDigits[#]] < 3 &]
```
Out[269]=
```
{2, 11, 31, 41, 61, 71, 101, 131, 151, 181, 191, 211,
 241, 251, 271, 281, 311, 331, 401, 421, 431, 461, 491, 521, 541}
```

In[270]:=
```
Select[RomanNumeral[Range[100]], ! MemberQ[Characters[#], "I"] &]
```
Out[270]=
```
{V, X, XV, XX, XXV, XXX, XXXV, XL, XLV,
 L, LV, LX, LXV, LXX, LXXV, LXXX, LXXXV, XC, XCV, C}
```

In[271]:=
```
Select[RomanNumeral[Range[1000]], StringReverse[#] == # &]
```
Out[271]=
```
{I, II, III, V, X, XIX, XX, XXX, L, C, CXC, CC, CCC, D, M}
```

In[272]:=
```
Select[IntegerName[Range[100]], StringTake[#, 1] == StringTake[StringReverse[#], 1] &]
```
Out[272]=
```
{nineteen, twenty-eight, thirty-eight, eighty-one,
 eighty-three, eighty-five, eighty-nine, ninety-seven}
```

In[273]:=
```
Select[TextWords[WikipediaData["Words"]], StringLength[#] > 15 &]
```
Out[273]=
```
{yibi-jarran-gabun, yibi-gabun-jarran, orthographically,
 multiple-morpheme, Proto-Indo-European, 978-0-08-044854-1}
```

In[274]:=
```
NestList[If[EvenQ[#], # / 2, 3 # + 1] &, 1000, 200]
```
Out[274]=
```
{1000, 500, 250, 125, 376, 188, 94, 47, 142, 71, 214, 107, 322, 161, 484, 242, 121, 364,
 182, 91, 274, 137, 412, 206, 103, 310, 155, 466, 233, 700, 350, 175, 526, 263,
 790, 395, 1186, 593, 1780, 890, 445, 1336, 668, 334, 167, 502, 251, 754, 377,
 1132, 566, 283, 850, 425, 1276, 638, 319, 958, 479, 1438, 719, 2158, 1079, 3238,
 1619, 4858, 2429, 7288, 3644, 1822, 911, 2734, 1367, 4102, 2051, 6154, 3077,
 9232, 4616, 2308, 1154, 577, 1732, 866, 433, 1300, 650, 325, 976, 488, 244, 122,
 61, 184, 92, 46, 23, 70, 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2,
 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4,
 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1,
 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2}
```

In[275]:=

```
WordCloud[Select[TextWords[WikipediaData["Computers"]], StringLength[#] == 5 &]]
```

Out[275]=

In[276]:=

```
Select[WordList[],
 StringTake[#, 3] == StringTake[StringReverse[#], 3] && # ≠ StringReverse[#] &]
```

StringTake: Cannot take positions 1 through 3 in "a".

StringTake: Cannot take positions 1 through 3 in "a".

StringTake: Cannot take positions 1 through 3 in "ad".

General: Further output of StringTake::take will be suppressed during this calculation.

Out[276]=

```
{despised, detected, detested, drainboard,
 foolproof, lackadaisical, marjoram, revolver}
```

Need a conditional test that the string has at least length 3 to avoid the errors.

In[277]:=
```
Select[WordList[], StringLength[#] == 10 && Total[LetterNumber[#]] == 100 &]
```

Out[277]=
```
{accumulate, alienation, answerable, apoplectic, aquamarine, bewitching, censurable,
 ceramicist, chastening, chimpanzee, clinically, collecting, condensate,
 congenital, conjugated, connivance, declension, deliquesce, demobilize,
 demodulate, denominate, diagonally, discipline, discommode, egoistical,
 emasculate, embodiment, emendation, empathetic, fatalistic, fatherhood,
 geographer, hemoglobin, inadequacy, interbreed, leveraging, liberalism,
 likelihood, martingale, mercantile, meridional, neoclassic, paramecium,
 plebiscite, potbellied, quadrangle, reciprocal, regimented, reschedule,
 researcher, scoreboard, septicemia, shibboleth, sleepyhead, stagecraft,
 stalemated, temperance, thickening, threatened, uncombined, unmodified}
```