

# Brian — PS 9 — 2025-02-21 — Solution

*EIWL3* Sections 23, 24, and 25

---

## Exercises from *EIWL3* Section 23

```
In[35]:= (* 23.1 *) N[Sqrt[2], 500]
```

```
Out[35]=
```

```
1.4142135623730950488016887242096980785696718753769480731766797379907324784621070·  
38850387534327641572735013846230912297024924836055850737212644121497099935831413·  
22266592750559275579995050115278206057147010955997160597027453459686201472851741·  
86408891986095523292304843087143214508397626036279952514079896872533965463318088·  
29640620615258352395054745750287759961729835575220337531857011354374603408498847·  
16038689997069900481503054402779031645424782306849293691862158057846311159666871·  
30130156185689872372
```

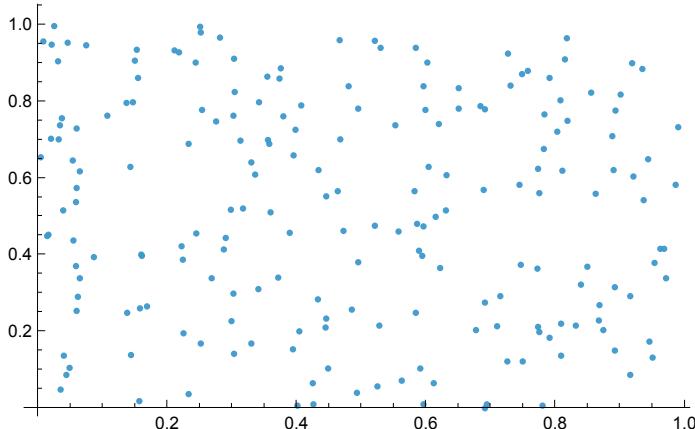
```
In[36]:= (* 23.2 *) RandomReal[1, 10]
```

```
Out[36]=
```

```
{0.920243, 0.0215262, 0.604632, 0.10347,  
0.380753, 0.853187, 0.904402, 0.539973, 0.425172, 0.910899}
```

```
In[37]:= (* 23.3 *) ListPlot[RandomReal[1, {200, 2}]]
```

```
Out[37]=
```



```
In[38]:= (* 23.4 *) Graphics[Line[AnglePath[RandomReal[2 Pi, 1000]]]]
Out[38]=
```



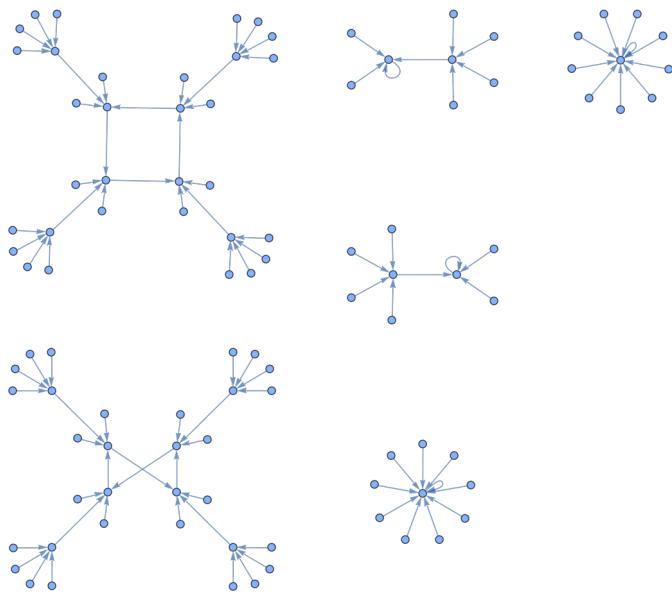
```
In[39]:= (* 23.5 *) Table[Mod[n^2, 10], {n, 0, 30}]
Out[39]= {0, 1, 4, 9, 6, 5, 6, 9, 4, 1, 0, 1, 4, 9, 6, 5, 6, 9, 4, 1, 0, 1, 4, 9, 6, 5, 6, 9, 4, 1, 0}
```

```
In[40]:= (* 23.6 *) Table[Mod[n^n, 10], {n, 100}]
Out[40]= {1, 4, 7, 6, 5, 6, 3, 6, 9, 0, 1, 6, 3, 6, 5, 6, 7, 4, 9, 0, 1, 4, 7, 6,
5, 6, 3, 6, 9, 0, 1, 6, 3, 6, 5, 6, 7, 4, 9, 0, 1, 4, 7, 6, 5, 6, 3, 6, 9,
0, 1, 6, 3, 6, 5, 6, 7, 4, 9, 0, 1, 4, 7, 6, 5, 6, 3, 6, 9, 0, 1, 6, 3, 6,
5, 6, 7, 4, 9, 0, 1, 4, 7, 6, 5, 6, 3, 6, 9, 0, 1, 6, 3, 6, 5, 6, 7, 4, 9, 0}
```

```
In[41]:= (* 23.7 *) Round[Table[Pi^i, {i, 10}]]
Out[41]= {3, 10, 31, 97, 306, 961, 3020, 9489, 29809, 93648}
```

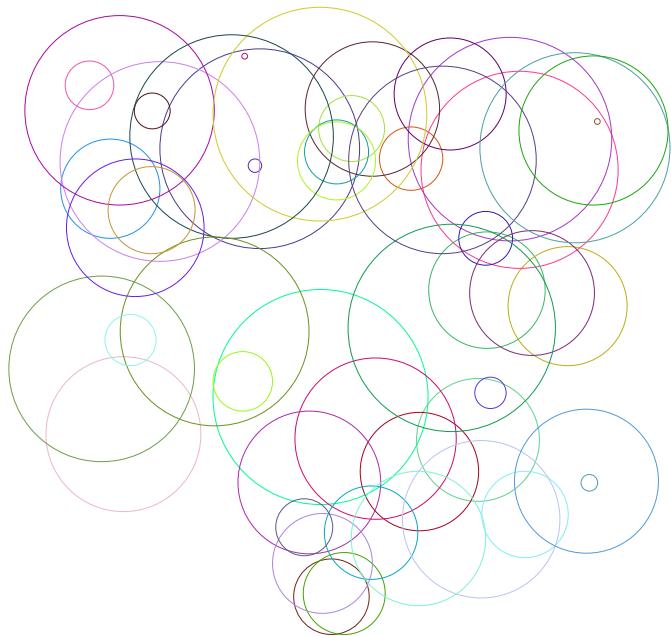
```
In[42]:= (* 23.8 *) Graph[Table[n → Mod[n^2, 100], {n, 0, 99}]]
```

Out[42]=



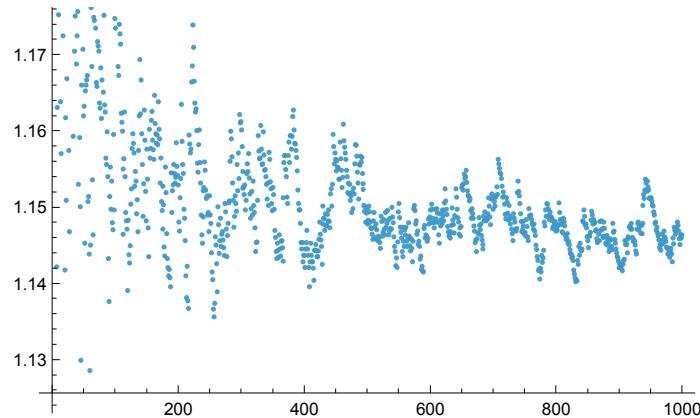
```
In[43]:= (* 23.9 *) Graphics[Table[
  Style[Circle[{RandomReal[10], RandomReal[10]}, RandomReal[2]], RandomColor[]],
  50]]
(* This is an expression that is just complicated enough that
I decided to use indenting to help me write it out correctly. *)
```

Out[43]=



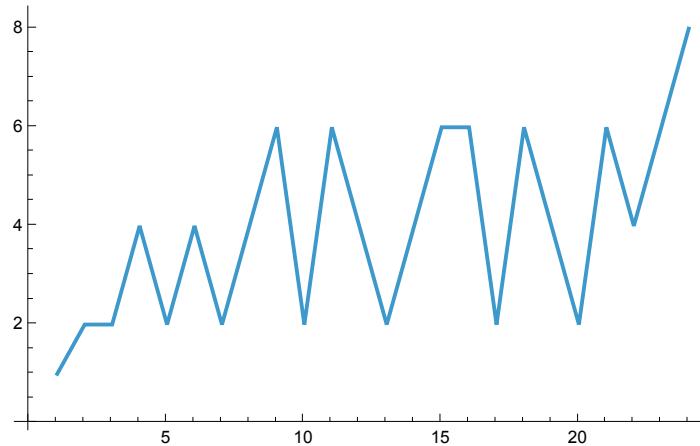
```
In[44]:= (* 23.10 *) ListPlot[Table[Prime[n]/(n Log[n]), {n, 2, 1000}]]
```

Out[44]=



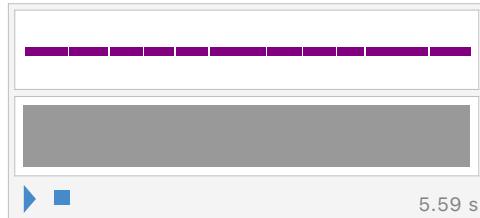
```
In[45]:= (* 23.11 *) ListLinePlot[Table[Prime[n] - Prime[n - 1], {n, 2, 25}]]  
(* I am not sure how we were supposed to know that the 25th prime was *)  
(* the last one less than 100. I just did trial and error to figure that out. *)
```

Out[45]=



```
In[46]:= (* 23.12 *) Sound[Table[SoundNote[0, RandomReal[0.5]], 20]]
```

Out[46]=

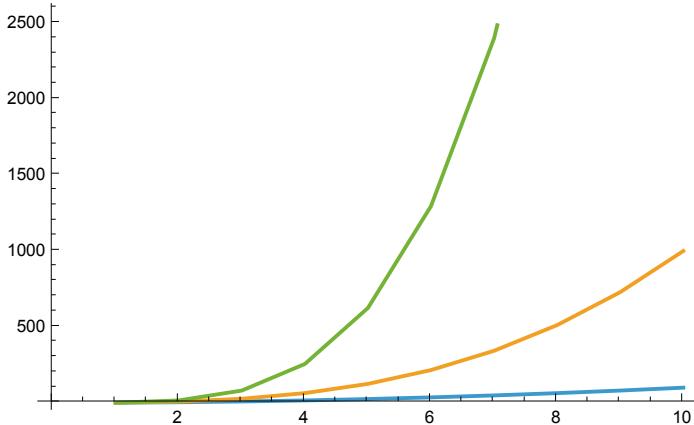


---

## Exercises from EWL3 Section 24

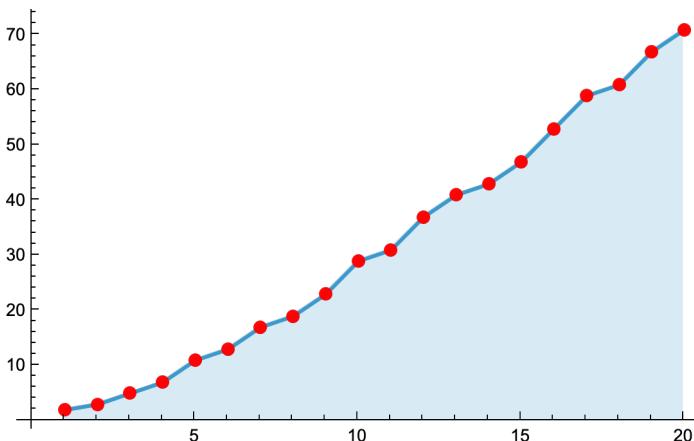
```
In[47]:= (* 24.1 *) ListLinePlot[{Range[10]^2, Range[10]^3, Range[10]^4}]
```

Out[47]=



```
In[48]:= (* 24.2 *) ListLinePlot[Table[Prime[i], {i, 20}],  
    Filling -> Axis, Mesh -> All, MeshStyle -> Red]
```

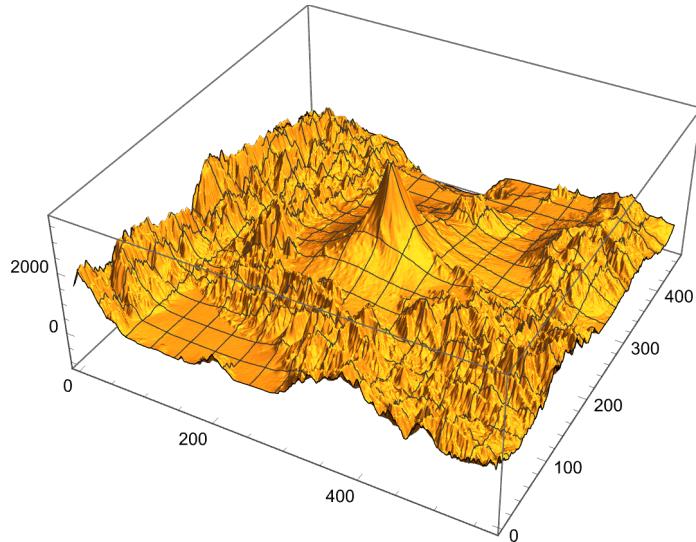
Out[48]=



In[49]:= (\* 24.3 \*)

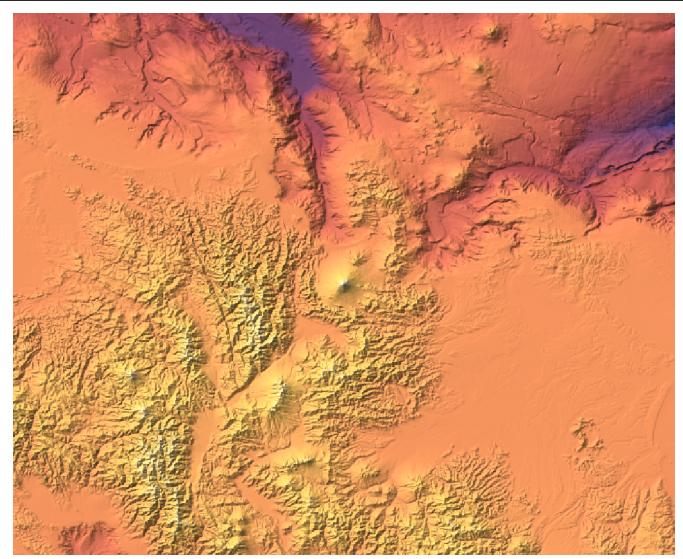
```
ListPlot3D[GeoElevationData[GeoDisk[Mount Fuji MOUNTAIN, 20 mi]], PlotRange -> All]
```

Out[49]=



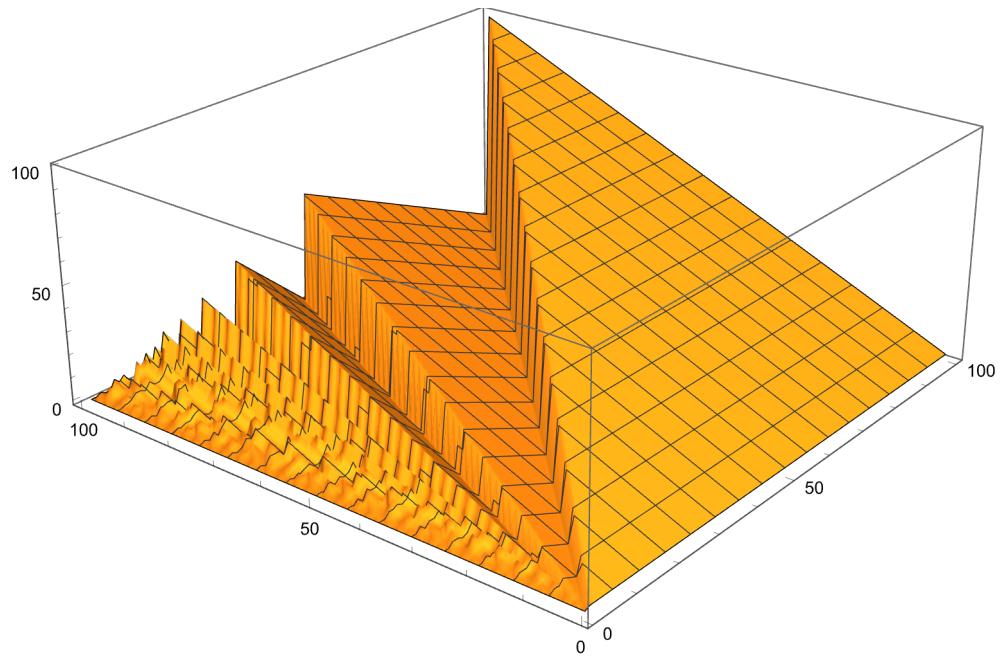
In[50]:= (\* 24.4 \*) ReliefPlot[GeoElevationData[GeoDisk[Mount Fuji MOUNTAIN, 100 mi]]]

Out[50]=



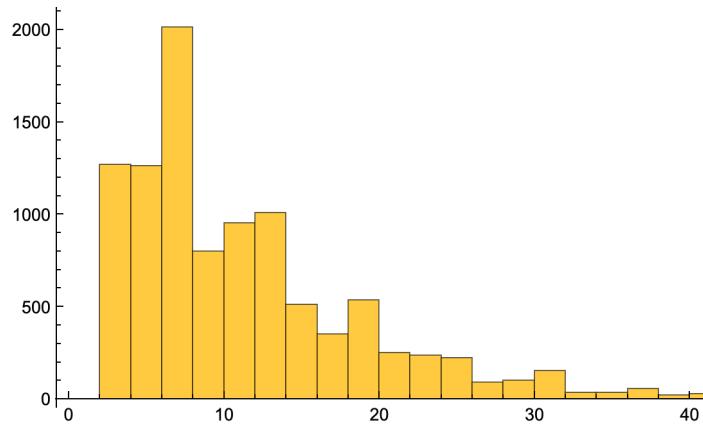
```
In[51]:= (* 24.5 *) ListPlot3D[Table[Mod[i, j], {i, 100}, {j, 100}]]
```

```
Out[51]=
```

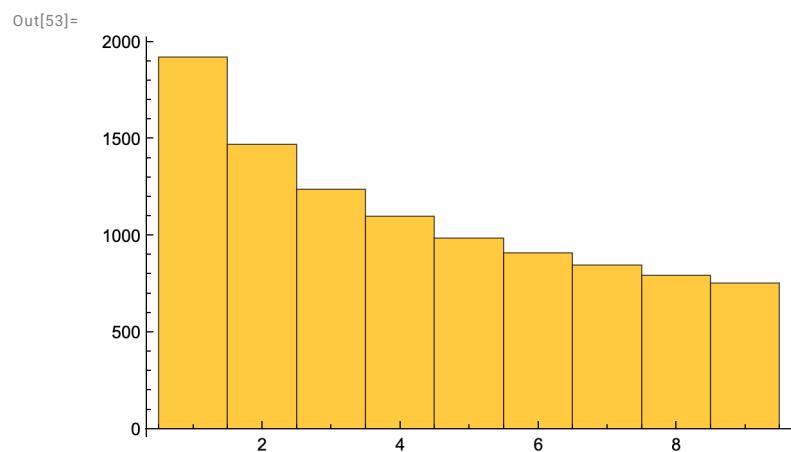


```
In[52]:= (* 24.6 *) Histogram[Table[Prime[j + 1] - Prime[j], {j, 9999}]]
```

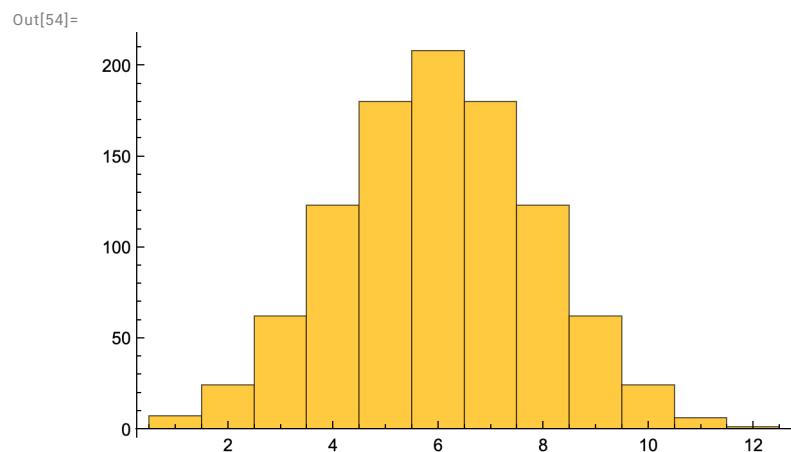
```
Out[52]=
```



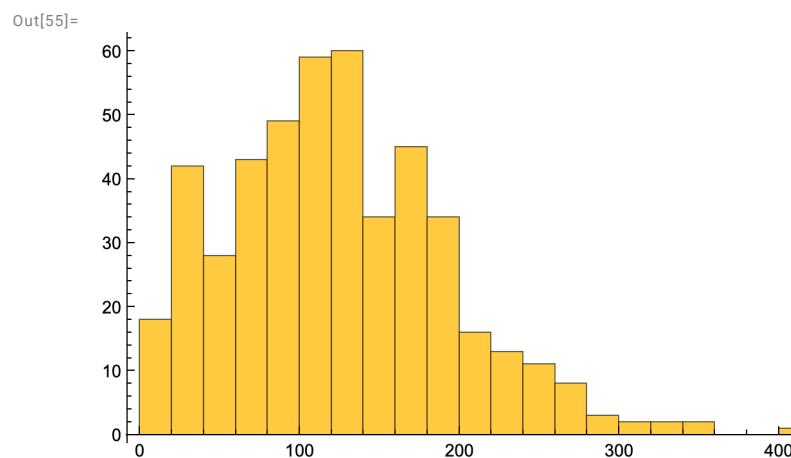
```
In[53]:= (* 24.7 *) Histogram[Table[First[IntegerDigits[j^2]], {j, 10000}]]
```



```
In[54]:= (* 24.8 *) Histogram[Table[StringLength[RomanNumeral[i]], {i, 1000}]]
```

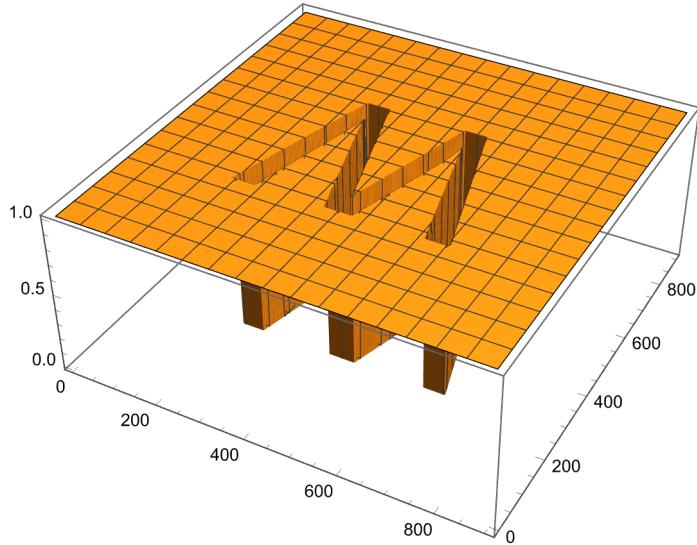


```
In[55]:= (* 24.9 *) Histogram[StringLength[TextSentences[WikipediaData["Computers"]]]]
```



```
In[56]:= (* 24.10 *) Table[Table[Total[RandomReal[100.0, n]], 2], {n, 1, 5}]
Out[56]= {{93.7468, 98.4759}, {110.828, 104.383},
{106.434, 150.809}, {201.268, 145.158}, {260.612, 147.252}}
```

```
In[57]:= (* 24.11 *) ListPlot3D[ImageData[Binarize[Graphics[Style[Text["W"], 200]]]]]
Out[57]=
```



## Exercises from EWL3 Section 25

```
In[58]:= (* 25.1 *) f /@ Range[5]
Out[58]= {f[1], f[2], f[3], f[4], f[5]}

In[59]:= (* This is indeed the same result as *) Table[f[n], {n, 5}]
(* We now eschew the bloated code we first learned. *)
(* This is going to eliminate a lot of our uses of Table[]. *)
Out[59]= {f[1], f[2], f[3], f[4], f[5]}

In[60]:= (* 25.2 *) f /@ g /@ Range[10]
Out[60]= {f[g[1]], f[g[2]], f[g[3]], f[g[4]],
f[g[5]], f[g[6]], f[g[7]], f[g[8]], f[g[9]], f[g[10]]}

In[61]:= (* 25.3 *) x // d // c // b // a
Out[61]= a[b[c[d[x]]]]
```

```
In[62]:= (* This is indeed the same result as *) a[b[c[d[x]]]]
Out[62]= a[b[c[d[x]]]]
```

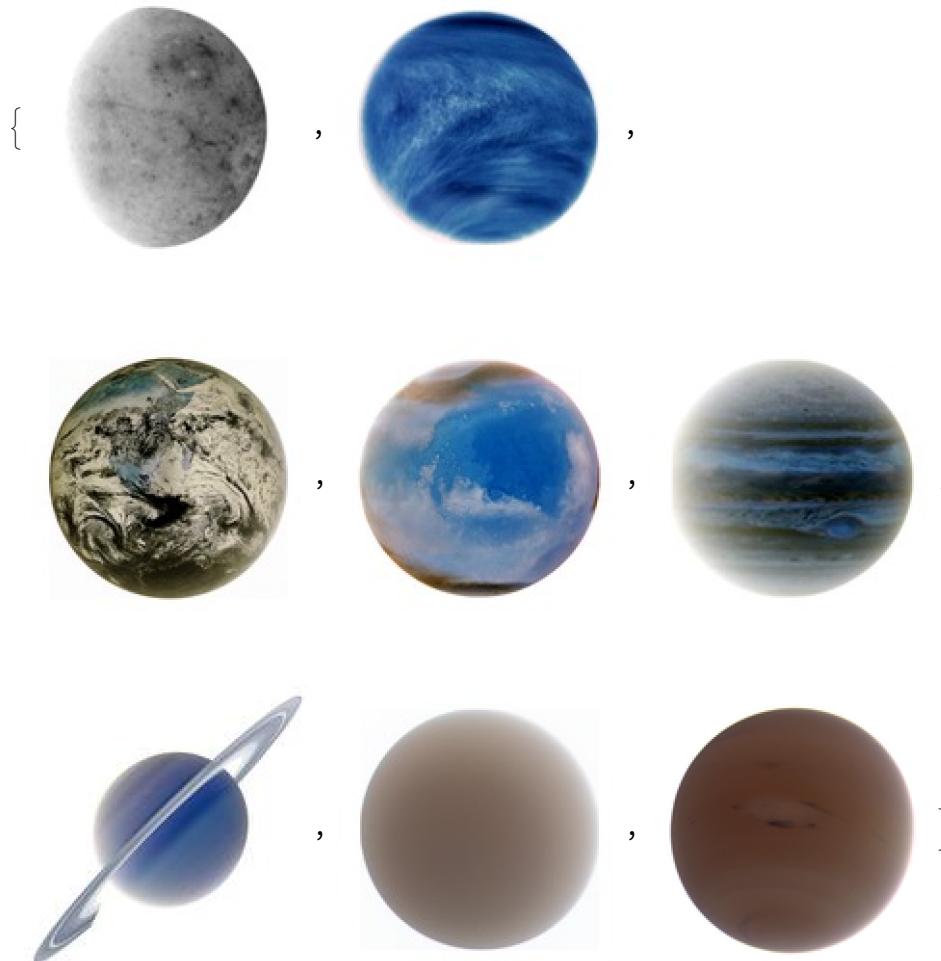
```
In[63]:= (* 25.4 *) Framed /@ Alphabet[]
```

Out[63]=

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

```
In[64]:= (* 25.5 *) ColorNegate /@  [  ]
```

Out[64]=



```
In[65]:= (* 25.6 *) GeoGraphics /@ EntityList[ Group of 5 COUNTRIES ]  
(* We learned about EntityList[] back in Section 16, *)  
(* but I will be frank and admit that I could not *)  
(* remember that I first needed to call EntityList[], *)  
(* and even more frankly, I'll admit that I am still *)  
(* unclear when EntityList[] and EntityValue are needed. *)
```

GeoServer: Unable to download one or more vector tiles.

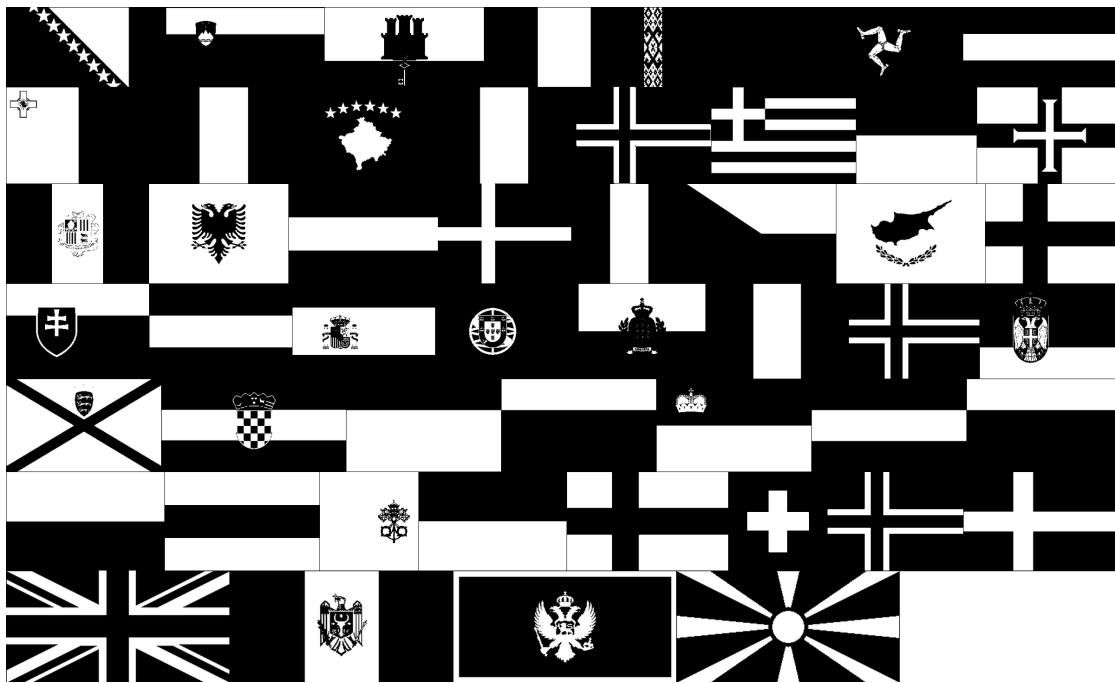
GeoServer: Unable to download one or more vector tiles.

Out[65]=



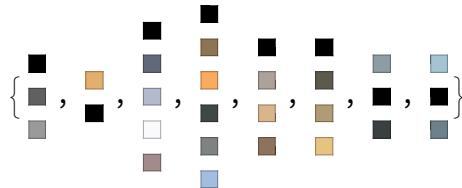
In[66]:= (\* 25.7 \*) Binarize /@ Europe COUNTRIES [ flag ] // ImageCollage

Out[66]=



In[67]:= (\* 25.8 \*) Column /@ DominantColors /@ planets PLANETS [ image ]

Out[67]=



In[68]:= (\* 25.9 \*) Total[LetterNumber["Wolfram"]]

Out[68]=

88