Generating random walks

(Sethna, "Entropy, Order Parameters, and Complexity", ex. 2.5)

Write a routine RandomWalk[N,d] to generate an N-step random walk in d dimensions, with each step uniformly distributed in (-1/2,1/2) in each dimension.
For d>1, generate the steps first as an Nxd array, and then do a cumulative sum with Accumulate.
Define d=1 separately, just generating N random numbers and then accumulating. (*Mathematica* treats an Nx1 array as a list of lists, each with one element.)

```
RandomWalk[N_, 1] := (steps = ...; Accumulate[steps])
RandomWalk[N_, d_] := (steps = ...; ...)
```

ListPlot some one dimensional random walks versus step number, for N=10000 steps. ListPlot some two-dimensional random walks with N=10000 steps.
Set AspectRatio→Automatic to make the x and y scales the same.

```
ListPlot[Table[RandomWalk[...], {i, 1, 3}], Joined → True]
```

```
ListPlot[Table[..., {i, 1, 3}], Joined → True, AspectRatio → Automatic]
```

Now write a routine Endpoints[W, N, d] that just returns the endpoints of W random walks of N steps each in d dimensions. (No need to use Accumulate; just use Total. If you generate a 3D array of size (N, W, d), Total will sum over the N steps of each walk. Again, define d=1 separately.

```
Endpoints[W_, N_, 1] := Total[...]
Endpoints[W_, N_, d_] := Total[...]
```

ListPlot the endpoints of 10000 random walks of length 10, together with the endpoints of 10000 random walks of length 1, appropriately setting the AspectRatio (and, of course, without Joined→True). Discuss.

```
ListPlot[{Endpoints[...], Endpoints[...]}, AspectRatio → Automatic]
```

Find σ for an N-step random walk with uniform step sizes in (-1/2,1/2). Compare the normalized histogram of 10000 endpoints with a normalized Gaussian of width σ, for N=1, 2, and 5.

```
hist =
 Histogram[{Endpoints[...], Endpoints[...], ...}, {0.1}, "ProbabilityDensity"]
Gauss[σ_] := (1 / Sqrt[...]) Exp[...]
σ[N_] := Sqrt[...]
Show[hist, Plot[Gauss[σ[1]], {x, -3 σ[1], 3 σ[1]}], Plot[...], ...]
```