A polygon is **convex** if any line segment joining any two vertices of the polygon stays inside the polygon. For example, a regular polygon is convex.

The **convex hull** of a set of points is the smallest convex polygon that contains all the points. The convex hull can be thought of by considering the points as nails sticking out of a board, and stretching an elastic band around the nails.

A common operation in graphics is to find the convex hull of a set of points. One way to draw the convex hull is to shade in the triangle formed by every triple of points. Here is code that does that.

```
shadeHull.m
```

But we really want to know the vertices on the outside (that is on the convex hull) and in what order. There are several common algorithms. We will do a simple one called **gift-wrapping** or **Jarvis march**. The idea is to find the bottommost point: that must be on the convex hull. Then think of attaching a string and winding the string around the points (say counter clockwise). The key task is to find the next point that the string will hit.

So given a `prevDir` of the previous line-segment found, we need to find the point whose angle from the current snag is closest in angle to `prevDir`. 

```
convexHull.m, drawConvexHull.m
```