#7: ROTATE(P, α) := $\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \cdot P$

Próbáljuk ki az új függvényt, az *A* csúcsot forgassuk el $\pi/6$-tal.

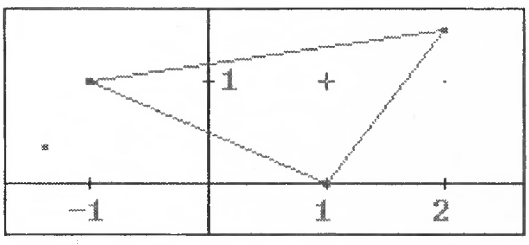
Írjuk be, egyszerűsítsük, majd értékeljük ki: *rotate(A,pi/6)*

#8: ROTATE$\left(A, \frac{\pi}{6}\right)$

#9: $\left[-\frac{\sqrt{3}}{2} - \frac{1}{2}, \frac{\sqrt{3}}{2} - \frac{1}{2}\right]$

#10: [-1.366025403, 0.3660254037]

Ábrázoljuk az elforgatott pontot a #10-el.



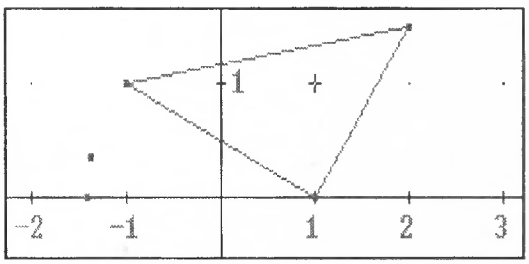
Próbáljuk ki a függvényt úgy is, hogy fokokban adjuk meg a szöget.

Írjuk be, egyszerűsítsük, majd értékeljük ki: *rotate(A,45°)*

#11: ROTATE(A, 45°)

#12: [-√2, 0]

Rajzoljuk be az új pontot.



Most a pont forgatására használt **ROTATION** függvényből építenünk kell egy *v* idomot forgató függvényt. Ugyanúgy, mint a háromszögnél, az idomot a csúcsainak vektoraival adjuk meg. A ROTATE_FIG definiálandó függvény a **ROTATE** függvényt használja, hogy az idom minden egyes csúcsát (azaz azok vektorelemeit) α szöggel elforgassa. A **VECTOR** függvényt arra használjuk, hogy az elforgatott pontokat egy új idommá állítsa össze.

Definiáljuk a ROTATE_FIG függvényt: *ROTATE_FIG(v,α):= vector(rotate(v↓i, α),i,1,dim(v))*

#13: ROTATE_FIG(v, α) := VECTOR(ROTATE(v[i], α), i, 1, DIM(v))