1. Somme de deux carrés

1. Ecrire une procédure testant si un entier $n \geq 1$ est somme de deux carrés, c'est-à-dire s'il est de la forme $i^2 + j^2$ avec i, j entiers naturels non nécessairement distincts.

2. Soit $\prod_i p_i^{\alpha_i}$ la décomposition de n en produit de facteurs premiers. Ecrire une procédure testant si :

$$\text{pour tout } i \text{ tel que } p_i \equiv 3 \ (4),\ \alpha_i \text{ est pair.} \tag{1}$$

(utiliser `ifactors`).

3. Un théorème de Fermat dit que n est somme de deux carrés si et seulement s'il satisfait la propriété (1). Vérifier ce théorème sur les entiers de 1 à 1000.

2. Courbes de poursuite

Une *courbe de poursuite* est la trajectoire suivie par un projectile qui se dirige à vitesse constante vers une cible en mouvement. Pour simplifier, on se place dans un plan et on se propose de programmer une simulation graphique.

On suppose que la cible $C = (X, Y)$ a une trajectoire d'équations paramétriques $X = f(t), Y = g(t)$, que le projectile $P = (x, y)$ est à la position (x_0, y_0) à l'instant $t = 0$ et que sa vitesse scalaire est v.

1. Ecrire une procédure `poursuite(f,g,x0,y0,v,T)` traçant simultanément les trajectoires suivies par C et P pendant l'intervalle de temps $[0, T]$ (indication : pour le projectile, choisir un pas de temps $h > 0$ et calculer de proche en proche sa position aux instants $h, 2h, 3h, \ldots$).

2. *Application 1 :* $f(t) = 0$, $g(t) = t$, $x_0 = 6$, $y_0 = 0$, $v = 2$, $T = 5$ (*courbe du chien*).

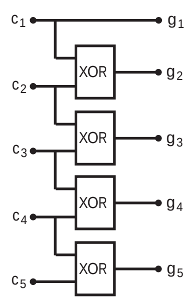
3. *Application 2 :* $f(t) = \cos 3t$, $g(t) = \sin 5t$, $x_0 = 6$, $y_0 = 0$, $v = 2$, $T = 10$.

3. Codes de Gray

1. Que fait la procédure suivante ?

```
> bin:=proc(n::nonnegint)
      if n<=1 then [n] else [op(bin(iquo(n,2))),irem(n,2)] end if
  end proc;
```

2. Le *code de Gray* $\overline{g_1 \ldots g_k}$ *de* n (sur k bits) s'obtient à partir du code binaire $\overline{c_1 \ldots c_k}$ grâce à un circuit composé de $k-1$ portes `XOR` disposées en parallèle, circuit représenté ci-contre pour $k = 5$. On rappelle que `XOR` est l'opérateur binaire *ou exclusif*, tel que a `XOR` $b = (a + b) \bmod 2$.



Ecrire à la main les codes de Gray des entiers de 0 à 7 (sur 3 bits).

3. En s'inspirant de la question 1, écrire une procédure qui renvoie le code de Gray d'un entier naturel quelconque.

4. Programmer la fonction $g : \mathbb{N} \to \mathbb{N}$ telle que $g(n)$ soit l'entier dont le code binaire est le code de Gray de n.

5. Représenter le graphe de g sur l'ensemble des entiers compris entre 0 et 1023.

6. Programmer la fonction réciproque $g^{-1} : \mathbb{N} \to \mathbb{N}$.

7. Expliquer pourquoi les codes de Gray de deux entiers consécutifs ne diffèrent que d'un seul bit *(c'est là tout l'intérêt !)*.