Solution du TP Maple n°2

1. Somme de deux carrés

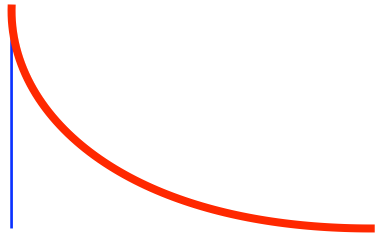
```
> restart:
> is_somme2carres:=proc(n::posint)
      # teste si n est somme de 2 carrés
      local C,k;
      C:=add(X^(k^2),k=0..isqrt(n));
      coeff(C^2,X,n)<>0;
  end:
> test:=proc(n)
     # teste la condition (1) de l'énoncé
     local L,i,p,a ;
     L:=ifactors(n)[2];
     for i to nops(L) do
       p:=L[i][1];
       a:=L[i][2];
       if irem(p,4)=3 and irem(a,2)=1 then return false end
  if;
     end do;
     return true
  end proc:
> # vérification du théorème pour les entiers de 1 à 1000 :
> L:=[$1..1000]:
> L1:=select(is_somme2carres,L):
> L2:=select(test,L):
> evalb(L1=L2);
```

true

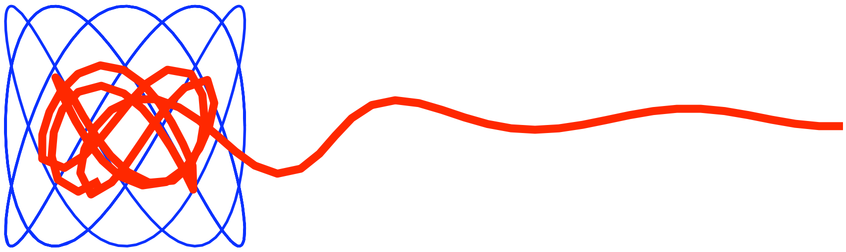
2. Courbes de poursuite

```
> restart:
> with(plots):
> poursuite:=proc(f,g,x0,y0,v,T)
      # trace les trajectoires de la cible et du projectile
      local t,h,dist,x,y,X,Y,P,cible,projectile;
      h:=evalf(T/100);
      x:=x0;y:=y0;P:=[x,y];dist:=1.0;
      for t from 0 by h to T while dist>0.02 do
        X:=f(t);Y:=g(t);
        dist:=sqrt((X-x)^2+(Y-y)^2);
        x:=x+v*h*(X-x)/dist;y:=y+v*h*(Y-y)/dist;
        P:=P,[x,y]
      end do;
      if dist<=0.02 then print(cat("Collision à l'instant ",
  convert(t,string))) end if;
      projectile:=plot([P],color=red,thickness=3);
      cible:=plot([f,g,0..t],color=blue);
      display([cible,projectile],axes=none)
  end:
> poursuite(t->0,t->t,6,0,2,5);
```

"Collision à l'instant 4.050000000"



```
> poursuite(t->cos(3*t),t->sin(5*t),6.,0.,2,10);
```



3. Codes de Gray

```
> restart:
> bin:=proc(n::nonnegint)
     # renvoie le code binaire de n (bit de poids fort à
  gauche)
     if n<=1 then [n] else [op(bin(iquo(n,2))),irem(n,2)] end
  if
  end proc:
> bin(10);
```

$$[1, 0, 1, 0]$$

```
> # comparaison avec la fonction prédéfinie de Maple :
> convert(10,base,2);
```

$$[0, 1, 0, 1]$$

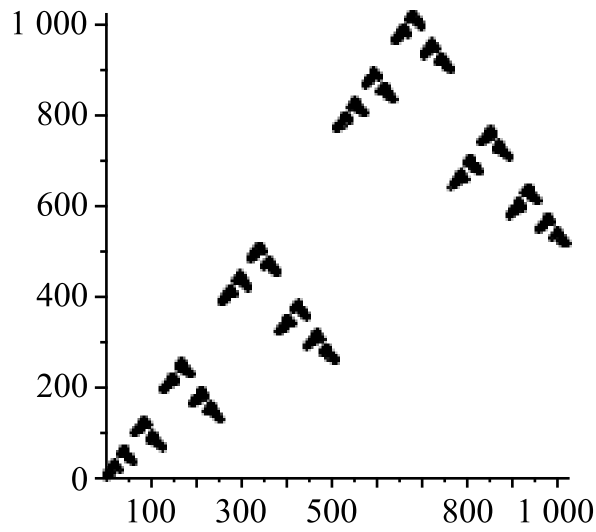
```
> gray:=proc(n::nonnegint)
     # renvoie le code de Gray de n
     local p;
     if n<=1 then [n] else p:=iquo(n,2);[op(gray(p)),irem(n+
  p,2)] end if
  end proc:
> gray(7);
```

$$[1, 0, 0]$$

```
> g:=proc(n::nonnegint)
     # la fonction g de l'énoncé
     local p;
     if n=0 then 0 else p:=iquo(n,2);2*g(p)+irem(n+p,2) end
  if
  end proc:
> g(2011);
```

$$1078$$

```
> # Graphe de g sur [0..1023] :
> L:=map(g,[$0..1023]):
> with(plots):
> listplot(L,style=point,scaling=constrained);
```




```
> ginv:=proc(n::nonnegint)
    # fonction inverse de g
    local p;
    if n=0 then 0 else p:=ginv(iquo(n,2));2*p+irem(p+n,2)
  end if
  end proc:
> ginv(1078);
```

2011

```
>
```

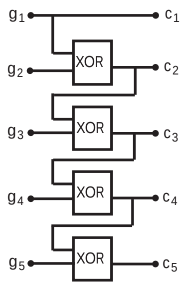
Réponses aux questions de l'exercice 3 :

1. La procédure `bin` renvoie, sous forme de liste, le code binaire de l'entier naturel n, avec les bits de poids fort à gauche.

2. Voici le tableau des codes de Gray des entiers de 0 à 7, ainsi que les valeurs de la fonction g :

n	binaire	Gray	$g(n)$
0	000	000	0
1	001	001	1
2	010	011	3
3	011	010	2
4	100	110	6
5	101	111	7
6	110	101	5
7	111	100	4

6. En notant $\oplus$ l'opérateur *ou exclusif*, pour $2 \leq i \leq k$ on a $g_i = c_{i-1} \oplus c_i$ donc $c_i = c_{i-1} \oplus g_i$. On en déduit que le codage de Gray est bijectif et que g est inversible. Voici d'ailleurs un circuit calculant le code binaire à partir du code de Gray (pour $k = 5$) :



7. Il suffit de prouver que les codes de Gray de deux entiers consécutifs $n, n+1$ diffèrent d'au plus un bit. La preuve se fait par récurrence sur n et résulte de l'écriture-même de la fonction `gray` :
- pour $n = 0$ c'est écrit,
- pour n pair de la forme $2p$, alors $\lfloor \frac{n}{2} \rfloor = \lfloor \frac{n+1}{2} \rfloor = p$ donc c'est évident,
- pour n impair de la forme $2p+1$, alors $\lfloor \frac{n}{2} \rfloor = p$ et $\lfloor \frac{n+1}{2} \rfloor = p+1$. Il faut montrer que dans ce cas, $(n + \lfloor \frac{n}{2} \rfloor) \bmod 2 = (n + 1 + \lfloor \frac{n+1}{2} \rfloor) \bmod 2$, i.e. $(2p + 1 + p) \bmod 2 = (2p + 2 + p + 1) \bmod 2$, ce qui est clair.