

TP8 : Entiers de Gauss et sommes de deux carrés

Eléments de corrigé

```
[ > restart;
```

Question 1

```
[ > znorm:=proc(z);  
  RETURN(z*conjugate(z));  
end;  
[ > znorm(1+I); znorm(1-I);  
2  
2
```

Question 2

```
[ > znormalize:=proc(z) local zz;  
  # renvoie le représentant modulo les inversibles qui est dans  
  le premier quadrant.  
  zz:=z;  
  if not zz=0 then while not(Re(zz)>0 and Im(zz)>=0) do  
    zz:=zz*I;  
  od;  
  fi;  
  RETURN(zz);  
end;  
[ > znormalize(1-I);  
1+I
```

Question 3

```
[ > zdiv:=proc(a,b) local x,y,q,r;  
  x:=Re(a/b);  
  y:=Im(a/b);  
  q:=round(x)+round(y)*I;  
  r:=a-q*b;  
  RETURN([q,r]);  
end;  
[ > res:=zdiv(7+I,4+3*I); (4+3*I)*res[1]+res[2];  
res:=[-1,2 I]  
7+I  
[ > res:=zdiv(4+3*I,1+I); (1+I)*res[1]+res[2];  
res:=[4-I,-1]  
4+3 I
```

Question 4

```
[ > zgcd:=proc(a,b) local za,zb,t;  
  za:=a;
```

```
  zb:=b;  
  t:=zdiv(za,zb);  
  while (t[2]<>0) do  
    za:=zb;  
    zb:=t[2];  
    t:=zdiv(za,zb);  
  od;  
  # le pgcd est stocké dans zb. On va le normaliser avant de  
  renvoyer le résultat.  
  RETURN(znormalize(zb));  
end;  
[ > zgcd(6,9);  
3  
[ > zgcd(7+I,4+3*I); zgcd(4+3*I,1+I);  
1  
1  
[ > a:=(1+I)*(4+3*I); b:=(1+I)*(7+I); zgcd(a,b);  
a:=1+7 I  
b:=6+8 I  
1+I
```

Question 5

```
[ > isprime(3); isprime(-3); isprime(6);  
true  
false  
false
```

à la différence de `isprime()` nous ne nous limitons pas aux représentants privilégiés des premiers ; de ce fait `iszprime(p)` renvoie vrai si et seulement si (p) est un idéal premier. On utilise alors le critère de primalité dans $\mathbb{Z}[i]$ énoncé dans la proposition 2. Les premiers normalisés (i.e. dans le premier quadrant) sont ceux de normes premières ou les entiers naturels premiers congrus à -1 modulo 4.

```
[ > iszprime:=proc(p) local zp;  
  zp:=znormalize(p);  
  if not(Im(zp)=0) then RETURN(isprime(znorm(zp)))  
  else  
    if isprime(zp) and mods(zp,4)=-1  
    then RETURN(true)  
    else RETURN(false);  
  fi;  
end;  
[ > iszprime(-3); iszprime(3*I);  
true  
true
```

Le résultat est cohérent (cf remarque précédente).

```
[ > iszprime(1+I); iszprime(1-I); iszprime(7+I); iszprime(4+3*I);  
true  
true
```

```

false
false
Pour la factorisation des entiers de Gauss de «petite norme», nous aurons besoin des nombres
premiers normalisés non entiers (i.e., ne provenant pas de  $\mathbb{Z}$ ) de norme  $\leq 25$ . Nous donnons
aussi la norme de ces premiers.
> L:={}: for a from 0 to 5 do
  for b from 0 to 5 do
    z:=a+b*I;
    if (znorm(z)<=25) and iszprime(z) then L:=L union
      {[znorm(z),znormalize(z)]}; fi;
  od;
od;
L;
{[13, 2+3 I], [17, 4+ I], [13, 3+2 I], [9, 3], [2, 1+ I], [17, 1+4 I], [5, 2+ I], [5, 1+2 I]}
> znorm(7+I);
50
On teste alors la divisibilité par des premiers de norme divisant 50 (donc  $\leq 25$ ) :
> zdiv(7+I,1+I);
[4-3 I, 0]
On poursuit avec 4-3I de norme 50/2=25 :
> zdiv(4-3*I,1+2*I);
[-2I, -I]
> zdiv(4-3*I,2+I);
[1-2 I, 0]
Et 1-2I=-I(2+I); finalement :
> -I*(1+I)*(2+I)^2;
7+I
> znorm(4+3*I);
25
> zdiv(4+3*I,1+2*I);
[2-I, 0]
> -I*(1+2*I)^2;
4+3 I
> znorm(5+3*I);
34
> zdiv(5+3*I,1+I);
[4-I, 0]
> -I*(1+I)*(1+4*I);
5+3 I
> znorm(7+2*I);
53
Or 53 est premier !
> iszprime(7+2*I);
true

```

Question 6

```
> randprime:=proc(n);
```

```

RETURN(prevprime(RandomTools[Generate](integer(range=5..n+1)
));
end;
c=a^k est d'ordre 4 si et seulement si  $c^2=-1$  modulo p; on fait varier aléatoirement le nombre
premier p et le a et note le nombre de cas favorables;
la procédure qui suit renvoie, après nba*nbp tentatives (nbp est le nombre de tirages de p et,
pour un p donné, nba le nombre de tirages de a), la probabilité de succès.
> teststrategie:=proc(nbp,nba) local i,j,k,p,a,c,nb;
  nb:=0;
  for i from 1 to nbp
  do
    p:=randprime(1000);
    while ((p-1) mod 4 <> 0)
    do
      p:=randprime(1000);
    od;
    k:=(p-1)/4;
    for j from 1 to nba
    do
      a:=RandomTools[Generate](integer(range=1..p-1));
      # noter qu'il vaudrait mieux prendre
      a:=RandomTools[Generate](integer(range=2..p-2));
      # car les classes de -1 et 1 ne sont pas d'ordre 4 !
      c:=a^k mod p;
      if mods(c&^2,p)=-1 then nb:=nb+1; fi;
    od;
  od;
  print(nb/(nba*nbp));
end;
> teststrategie(1000,1); teststrategie(100,10);
123
250
533
1000

```

On obtient un résultat très proche de 1/2.

Question 7

```
> for i from 1 to 10 do teststrategie(1,100); od;
11
25
27
50
51
100
23
50
49
100
```

41
100
57
100
21
50
13
25
12
25

Pour ces dix choix aléatoires de p, la probabilité (à p fixé) d'obtenir un «bon» a est proche de 1/2.

Question 8

```
> ordre4:=proc(p) local k,a,c;
  if not(isprime(p) and ((p-1) mod 4=0))
  then RETURN(FAIL);
  else
    k:=(p-1)/4;
    a:=RandomTools[Generate](integer(range=2..p-2));
    c:=mods(a&^k,p);
    while not(mods(c^2,p)=-1)
    do
      a:=RandomTools[Generate](integer(range=2..p-2));
      c:=mods(a&^k,p);
    od;
    RETURN(c);
  fi;
end:
> ordre4(13);
-5
> ordre4(997);
-161
```

Question 10

```
> DeuxCarres:=proc(p) local a,b,c,d;
  if (not(isprime(p)))
  then error "Le nombre %1 n'est pas premier",p;
  else
    if (p mod 4 <>1)
    then error "Le nombre premier %1 n'est pas de la
forme 4k+1",p;
    else
      c:=ordre4(p);
      d:=zgcd(p,I-c);
      a:=Re(d);
      b:=Im(d);
      RETURN([a,b,a^2+b^2]);
    fi;
  fi;
end;
```

```
# la dernière composante est là uniquement pour vérification
  fi;
end:
> DeuxCarres(2); DeuxCarres(9); DeuxCarres(997);
DeuxCarres(13);
Error, (in DeuxCarres) Le nombre premier 2 n'est pas de la forme 4k+1
Error, (in DeuxCarres) Le nombre 9 n'est pas premier
[6, 31, 997]
[3, 2, 13]
> DeuxCarres(1000117);
[46, 999, 1000117]
> DeuxCarres(281474976710677);
[15398649, 6660074, 281474976710677]
```

Question 11

Ce qu'on a fait à la main sur des exemples à la question 5 se généralise à un entier de Gauss z quelconque, modulo le fait que l'on n'a pas à disposition cette liste d'irréductibles parmi lesquels piocher. On prend un diviseur premier p de $N(z)$. Si p est congru a -1 modulo 4, c'est un facteur irréductible cherché. Si par contre p est congru à 1 modulo 4, il s'écrit $p=a^2+b^2$. Alors $a+ib$ ou $a-ib$ divise z, et l'on recommence avec le quotient de z par le diviseur obtenu. A implémenter si le coeur vous en dit !