Le puzzle de Dudeney et le triangle récalcitrant

Renée De Graeve

10 mai 2020

1 Avant propos

On écrit, tout d'abord les programmes qui permettent de réaliser le puzzle de Henry Dudeney (1857-1930) : on découpe en 4 morceaux un triangle et avec ces 4 morceaux on peut reconstituer un carré ou un rectangle (cf `La classe de 6ième et le puzzle de Dudeney`).
Mais il existe des triangles pour lesquels la méthode de Dudeney ne marche pas !
Ce sera ensuite, l'objet de l'exercice proposé : on prend comme exemple un triangle nommé "le triangle récalcitrant" et on vous propose de le découper en un minimum de morceaux, pour pouvoir, avec ces morceaux, reconstituer un carré.
Il y a plusieurs découpages avec 5 morceaux mais je n'en ai pas trouvé avec 4.

2 Le puzzle de Dudeney transformant un triangle en un un carré

Soit le triangle de sommets dont les affixes sont $z1, z2, z3$.
On tape :

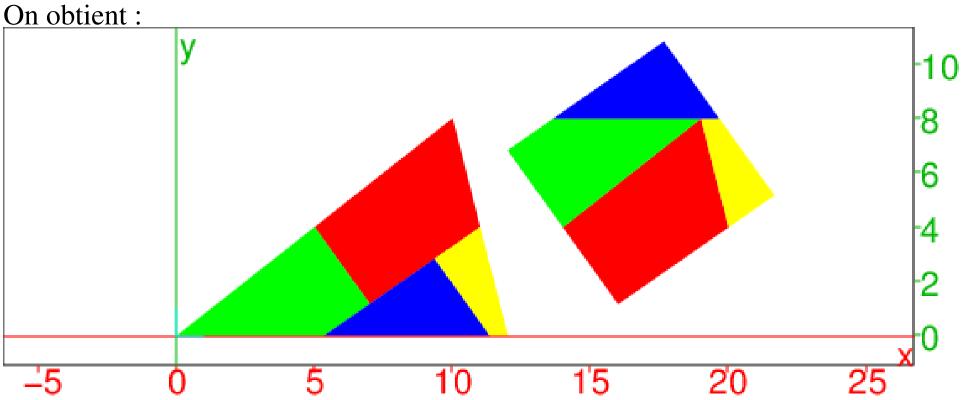
```
dudeneyc(z1,z2,z3):={
  local L1,L2,T1,T2,T3,F0,M,N,E,F,E1,M1,k,a,h,P1,P2,P3,P4,
        Q1,Q2,Q3,Q4;
  L1:=NULL;L2:=NULL;
  T1:=point(z1);
  T2:=point(z2);
  T3:=point(z3);
  M:=milieu(T1,T2);N:=milieu(T1,T3);a:=longueur(M,N);
  F0:=projection(droite(T2,T3),M);
  h:=longueur(M,F0);
  k:=a-sqrt(2*h*a-h^2);
  F:=k*(N-M)/a+F0;
  E:=(N-M)+F;
  E1:=projection(droite(F,N),E);
  M1:=projection(droite(F,N),M);
  P1:=polygone(T1,M,M1,N);
  P2:=polygone(M,M1,F,T2);
  P3:=polygone(E,T3,N,E1);
  P4:=polygone(E1,E,F);
```

```
  Q1:=translation(3*(z2+z3)/4,P1);
  Q2:=translation(3*(z2+z3)/4,symetrie(M,P2));
  Q3:=translation(3*(z2+z3)/4,symetrie(N,P3));
  Q4:=translation(3*(z2+z3)/4+2*(N-E),P4);
  L1:=L1,affichage(P1,1+rempli),affichage(P2,2+rempli),
        affichage(P3,3+rempli),affichage(P4,4+rempli);
  L2:=L2,affichage(Q1,1+rempli),affichage(Q2,2+rempli),
        affichage(Q3,3+rempli),affichage(Q4,4+rempli);
  retourne L1,L2;
}:;
```

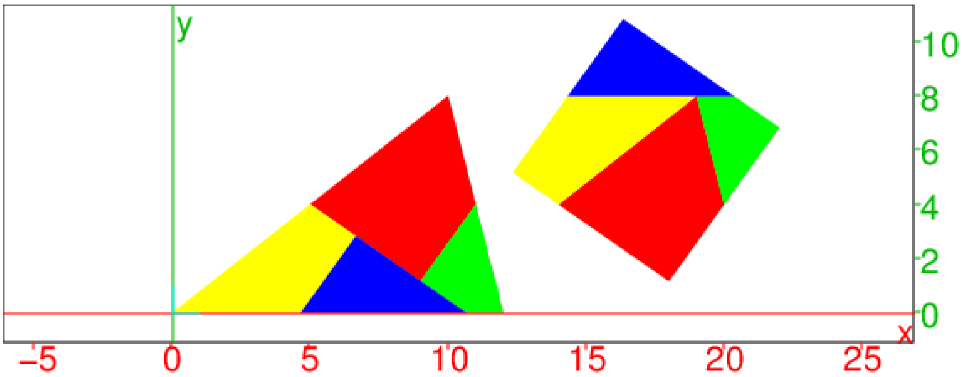
On peut donc avoir de 0 à 6 puzzles différents pour un triangle de sommets dont les affixes sont $z1, z2, z3$ car on peut utiliser :
`dudeneyc(z1,z2,z3)` ou
`dudeneyc(z1,z3,z2)` ou
`dudeneyc(z2,z3,z1)` ou
`dudeneyc(z2,z1,z3)` ou
`dudeneyc(z3,z2,z1)` ou
`dudeneyc(z3,z1,z2)`
Mais on peut avoir des triangles récalcitrants comme par exemple le triangle(18+12*i,0,12) pour lequel la méthode de Dudeney ne marche pas ou avoir des triangles qui auront 6 puzzles différents avec la méthode de Dudeney comme par exemple le triangle(10+8*i,0,12)

3 Exemple : les 6 puzzles de Dudeney pour le triangle(10+8*i,0,12)

On tape :
`dudeneyc(10+8*i,0,12)`
On obtient :



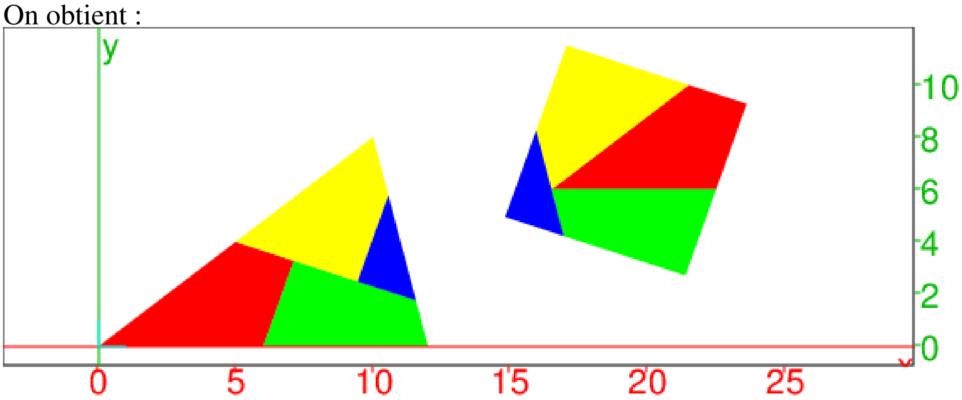
On tape :
`dudeneyc(10+8*i,12,0)`
On obtient :



On tape :

```
dudeneyc(0,12,10+8*i)
```

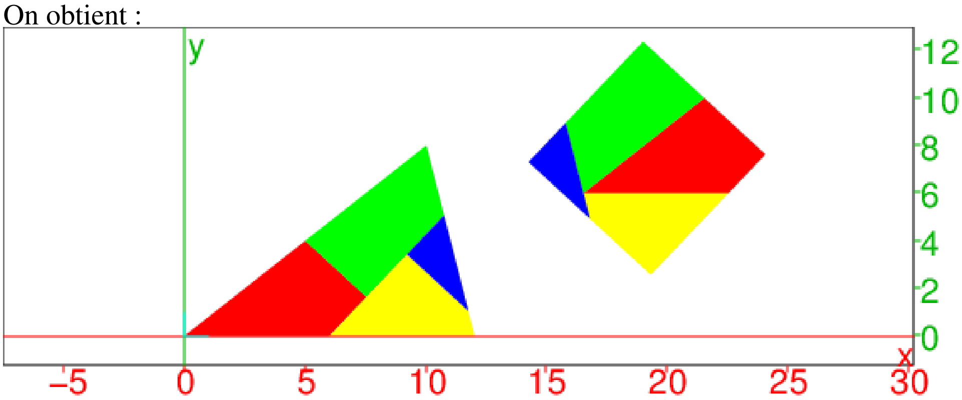
On obtient :



On tape :

```
dudeneyc(0,10+8*i,12)
```

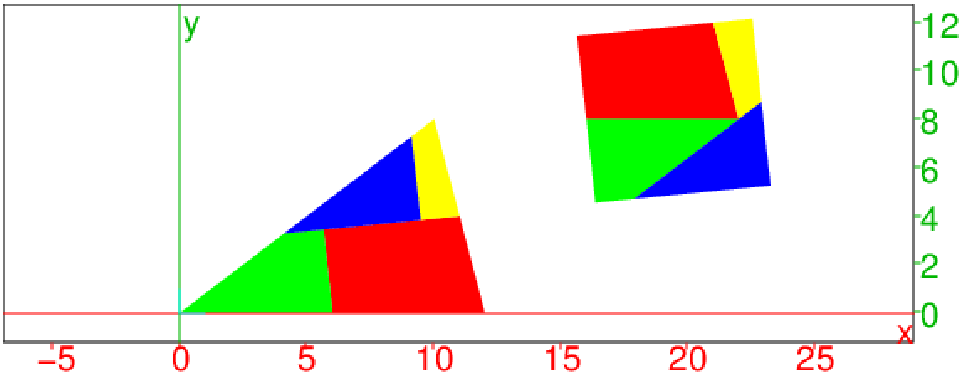
On obtient :



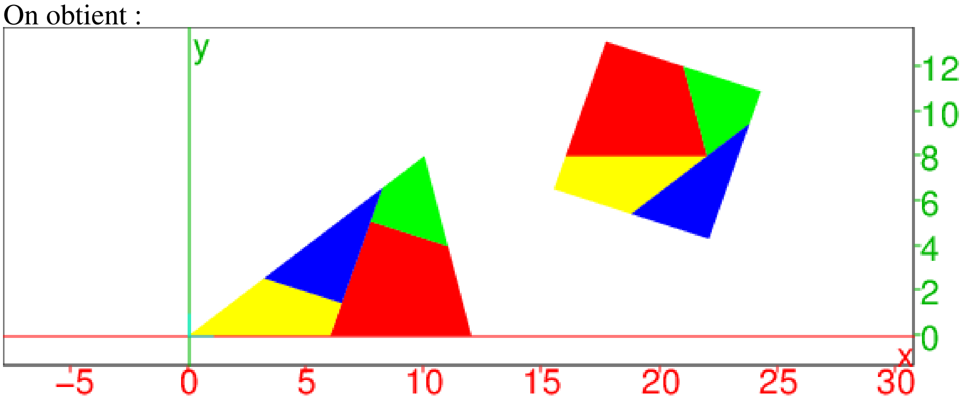
On tape :

```
dudeneyc(12,0,10+8*i)
```

On obtient :



On tape :
`dudeneyc(12,10+8*i,0)`
On obtient :



4 Le puzzle de Dudeney transformant un triangle en un rectangle

On peut aussi transformer un triangle en un tectangle dont un côté vaut l.
On tape pour que le rectangle formé par les 4 pièces soit translaté de $d*(z2+z3)$ (selon les cas $d=3/4$ ou $d=1$) :

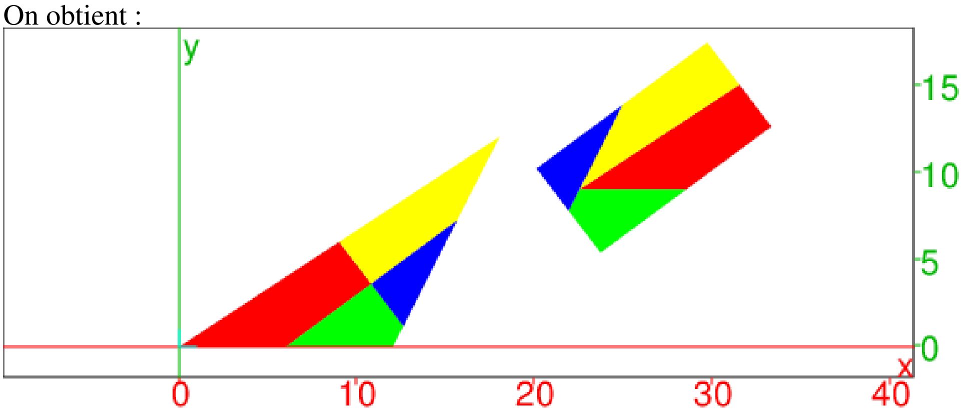
```
dudeneyr(z1,z2,z3,l,d):={
  local L1,L2,T1,T2,T3,F0,z0,M,N,E,F,z,E1,M1,k,a,h,c,
        P1,P2,P3,P4,Q1,Q2,Q3,Q4;
  L1:=NULL;L2:=NULL;
  T1:=point(z1);
  T2:=point(z2);
  T3:=point(z3);
  M:=milieu(T1,T2);
  N:=milieu(T1,T3);
  a:=longueur(M,N);
  F0:=projection(droite(T2,T3),M);
  z0:=affixe(F0);
  h:=longueur(M,F0);
  c:=2*longueur(T2,F0);
  si ((h-l)>0) alors return afficher("il faut",l>=h);fsi;
```

```
  k:=a-sqrt(l^2-h^2);
  z:=z0+(z3-z2)*k/(2*a);
  F:=point(z);E:=(N-M)+F;
  si (angle(E,T2,T3)==0) ou (angle(F,T2,T3)==0)  alors
      return afficher("l est trop grand ou trop petit");
  fsi;
  E1:=projection(droite(F,N),E);
  M1:=projection(droite(F,N),M);
  P1:=polygone(T1,M,M1,N);
  P2:=polygone(M,M1,F,T2);
  P3:=polygone(E,T3,N,E1);
  P4:=polygone(E1,E,F);
  Q1:=translation(d*(z2+z3),P1);
  Q2:=translation(d*(z2+z3),symetrie(M,P2));
  Q3:=translation(d*(z2+z3),symetrie(N,P3));
  Q4:=translation(d*(z2+z3)+2*(N-E),P4);
  L1:=L1,affichage(P1,1+rempli),affichage(P2,2+rempli),
         affichage(P3,3+rempli),affichage(P4,4+rempli);
  L2:=L2,affichage(Q1,1+rempli),affichage(Q2,2+rempli),
         affichage(Q3,3+rempli),affichage(Q4,4+rempli);
  retourne [L1,L2],z;
}:;
```

On tape :

```
dudeneyr(0,12,18+12*i,6,3/4)[0]
```

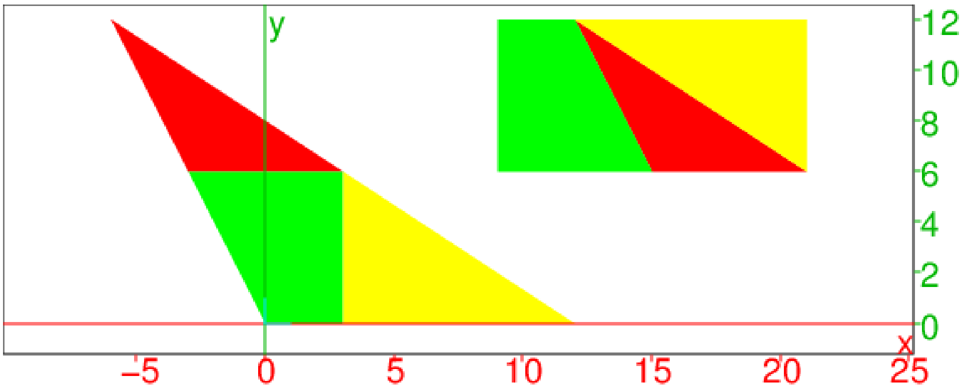
On obtient :



On tape :

```
dudeneyr(-6+12*i,0,12,6,3/2)[0]
```

On obtient :



5 Exemple d'un triangle récalcitrant

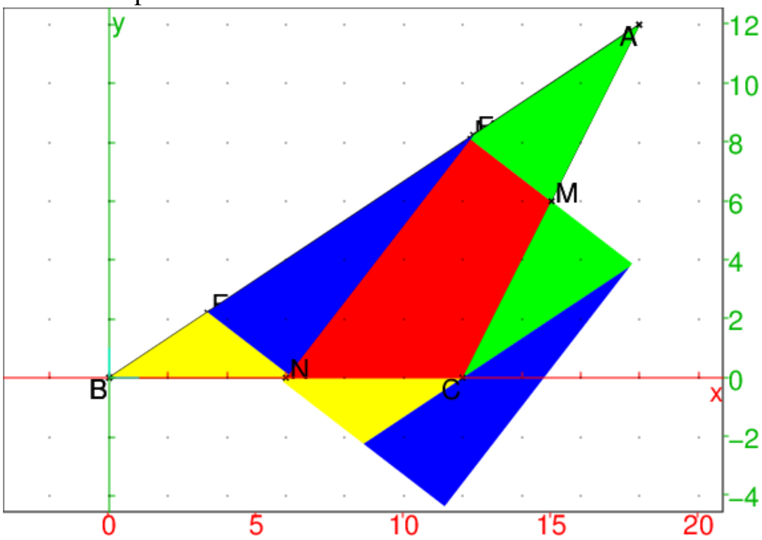
On considère le triangle ABC définit par :
`A:=point(18+12*i);B:=point(0);C:=point(12)`
On veut transformer ce triangle ABC en un carré avec la méthode de Dudeney.

5.1 Essai avec la méthode de Dudeney

Dans un écran de géométrie, on tape :

```
A:=point(18+12*i);B:=point(0);C:=point(12);
triangle(A,B,C);
H:=projection(droite(A,B),C);
M:=milieu(A,C);N:=milieu(B,C);
supposons(c=[0.97,-4.83,0.97,0.01]);
F0:=projection(droite(A,B),M);
E0:=projection(droite(A,B),N);
F:=translation((B-A)*c/5/sqrt(13),F0);
E:=translation((B-A)*c/5/sqrt(13),E0);
E1:=projection(droite(F,N),E);
M1:=projection(droite(F,N),M);
segment(F,N);
P1:=polygone(C,M,M1,N):; affichage(P1,1+rempli);
P2:=polygone(A,F,M1,M):; affichage(P2,2+rempli);
P3:=polygone(B,N,E1,E):; affichage(P3,3+rempli);
P4:=polygone(F,E1,E):; affichage(P4,4+rempli);
affichage(symetrie(M,P2),2+rempli);
affichage(symetrie(N,P3),3+rempli);
affichage(translation(2*(N-E),P4),4+rempli);
A:=A;M:=M;
evalf(longueur2(F,N));
F:=F;N:=N;
```

On obtient pour `c=0.8` :



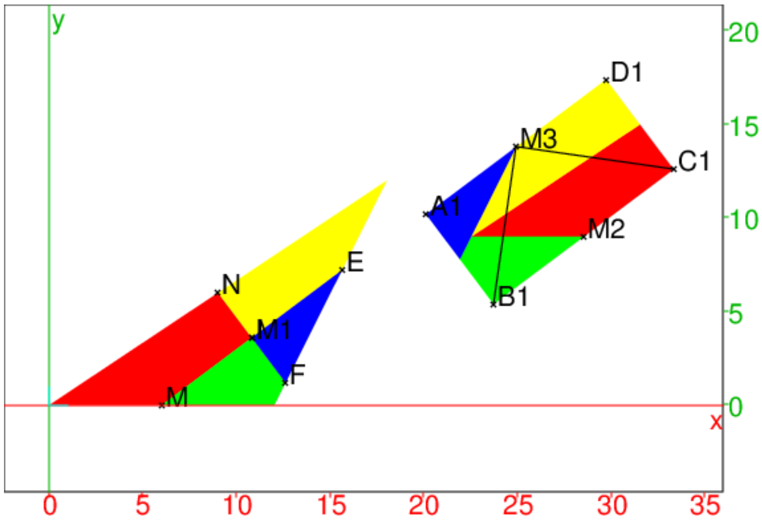
On peut faire varier `c` entre -4.83 et 0.97 mais cela transforme le triangle récalcitrant en un rectangle et on n'obtient jamais un carré.

6 Transformation du triangle récalcitrant en un carré

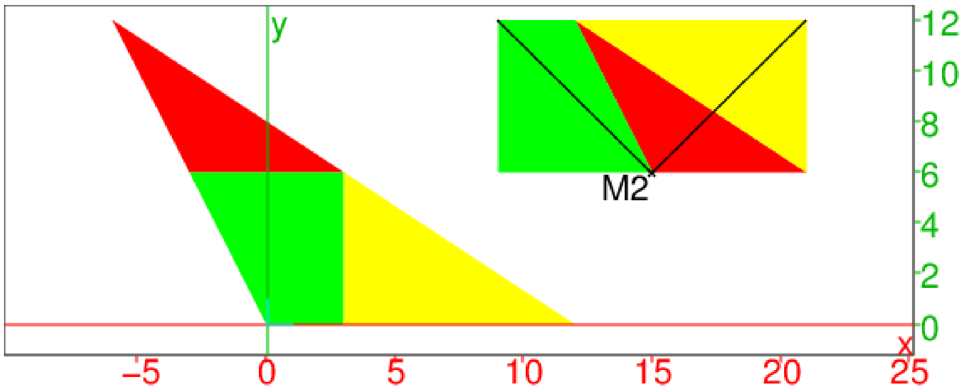
On veut transformer le triangle(-6+12*i,0,8) (ou le triangle(18+12*i,0,12) son symétrique) en un carré à l'aide d'un puzzle ayant le plus petit nombre n de pièces.
Question ouverte : Existe-t il une solution pour $n = 4$?
Voici dans les sections suivantes plusieurs façons de décomposer ce triangle en 5 pièces pour le transformer en carré.

6.1 On transforme le triangle en un rectangle longueur l et largeur $l/2$

Le triangle(18+12*i,0,12) (ou le triangle(-6+12*i,0,8)) a pour aire 72 donc le rectangle aura pour longueur $l = 12$ et comme largeur $l/2 = 6$.
On tape :
`dudeneyr(0,12,18+12*i,6,3/4)`
On obtient le losange $NMFE$ et 3+3+2+1=9 pièces pour le puzzle :



On tape :
`dudeneyr(-6+12*i,0,12,6)`
On obtient 6 pièces pour le puzzle :



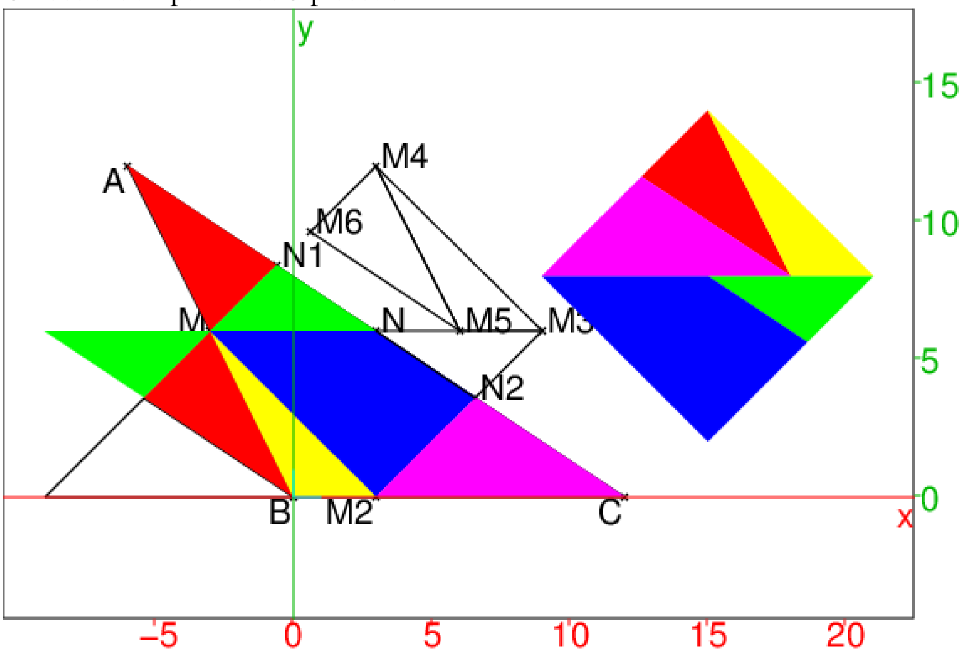
6.2 On transforme le triangle récalcitrant en un parallélogramme

On transforme le triangle ABC en le parallélogramme $BCND$.
On tape :

```
A:=point(-6+12*i);
B:=point(0);
C:=point(12);
triangle(A,B,C);
M:=affichage(milieu(A,B),quadrant2);
N:=milieu(A,C);
M2:=point(3);
N1:=inter_unique(droite(A,C),droite(y=x+9));
N2:=inter_unique(droite(A,C),droite(y=x-3));
M3:=symetrie(N,M);M4:=symetrie(N,M2);
M5:=translation((6+6*i),B);
M6:=translation((M-N1),M4)
```

```
P1:=polygone(M,N1,A):;Q1:=symetrie(M,P1):;
P2:=polygone(M,N,N1):;Q2:=symetrie(M,P2):;
P3:=polygone(M,B,M2):;Q3:=P3:;
P4:=polygone(M,M2,N2,N):;Q4:=P4:;
P5:=polygone(N2,M2,C):;Q5:=P5:;
translation((B-C),P5);
affichage(P1,1+rempli);affichage(Q1,1+rempli);
affichage(P2,2+rempli);affichage(Q2,2+rempli);
affichage(P3,3+rempli);
affichage(P4,4+rempli);
affichage(P5,5+rempli);
T1:=symetrie(milieu(N1,M6),P1):;T1;
T2:=symetrie(N,P2):;T2;
T3:=translation((M4-M),P3):;T3
T4:=P4:;
T5:=translation((M-M2),P5):;
affichage(translation(12+2i,T1),1+rempli);
affichage(translation(12+2i,T2),2+rempli);
affichage(translation(12+2i,T3),3+rempli);
affichage(translation(12+2i,T4),4+rempli);
affichage(translation(12+2i,T5),5+rempli);
```

On obtient un puzzle de 5 pièces :



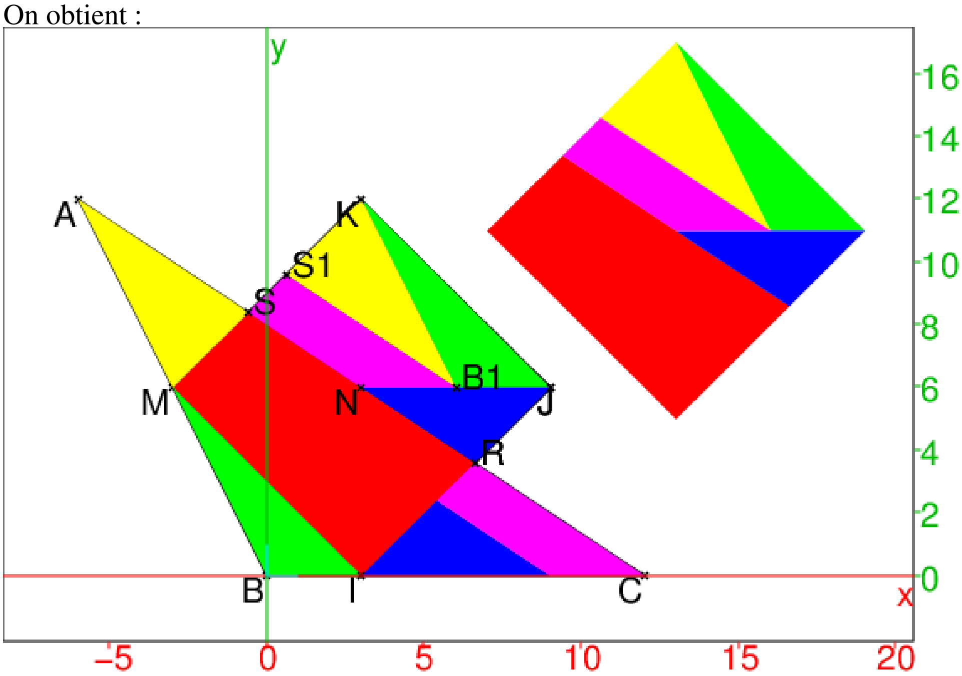
6.3 On positionne le carré comme à la section précédente

On tape :

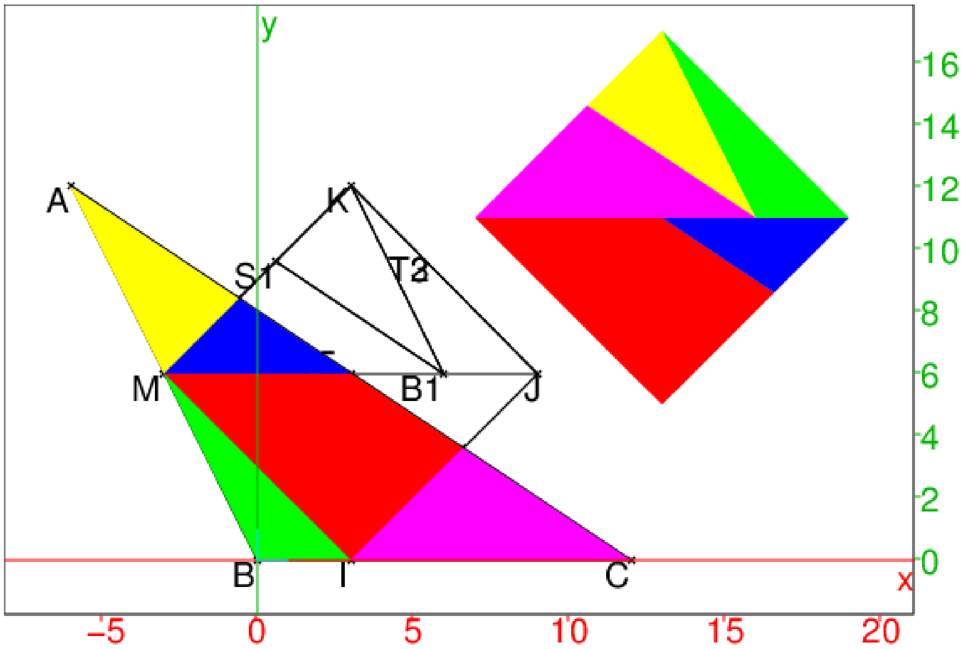
```
A:=point(-6+12*i);B:=point(0);C:=point(12);
triangle(A,B,C);
```

```
M:=point(-3+6*i);N:=point(3+6*i);I:=point(3);
carre(M,I,J,K);
R:=inter_unique(droite(A,C),droite(I,J));
S:=inter_unique(droite(A,C),droite(M,K));
P1:=polygone(M,I,R,S)::;Q1:=P1:;
P2:=polygone(M,I,B):;
Q2:=translation((K-M),P2):;
P3:=polygone(A,M,S):;
Q3:=symetrie(milieu(M,K),P3):;
Q4:=polygone(N,R,J):;
P4:=symetrie(milieu(I,J),Q4):;
B1:=translation((K-M),B);
S1:=symetrie(milieu(M,K),S);
Q5:=polygone(S,N,B1,S1):;
P5:=translation(R-S1,Q5):;
affichage(P1,1+rempli);affichage(P2,2+rempli);
affichage(P3,3+rempli);affichage(P4,4+rempli);
affichage(P5,5+rempli);
affichage(Q1,1+rempli);affichage(Q2,2+rempli);
affichage(Q3,3+rempli);affichage(Q4,4+rempli);
affichage(Q5,5+rempli);
```

On obtient :



alors que précédemment on avait :

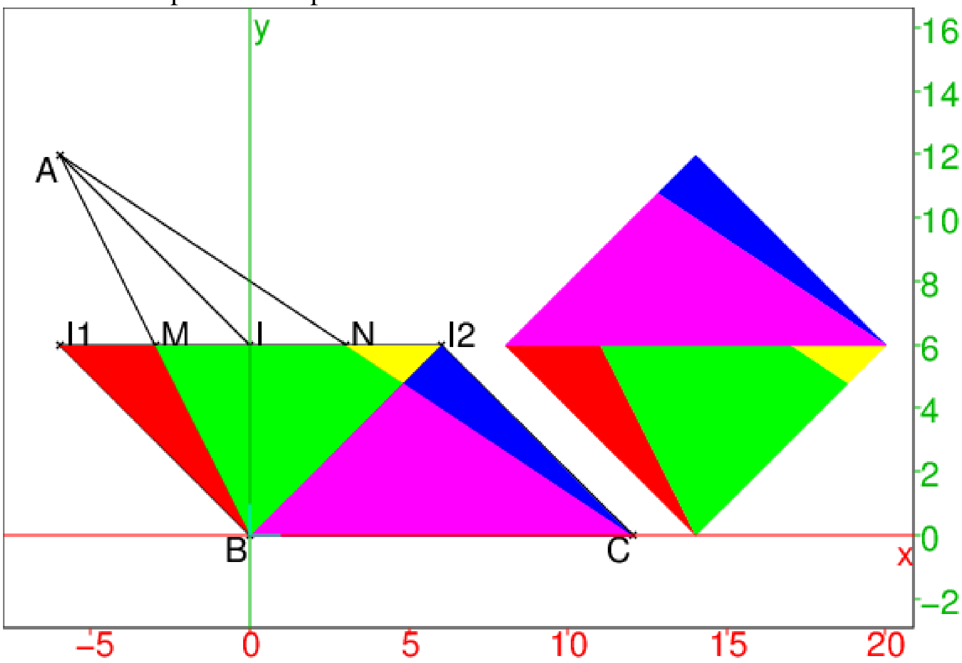


6.4 Avec un autre parallélogramme

On transforme le triangle ABC en le parallélogramme BCI_2I_1 avec `I1:=point(-6+6i)`.

```
A:=point(-6+12*i);B:=point(0);C:=point(12);
triangle(A,B,C);
M:=milieu(A,B);N:=milieu(A,C);
I:=milieu(M,N);segment(A,I);
I1:=symetrie(M,I);I2:=symetrie(N,I);
s:=segment(B,I2);polygone(I2,I1,B,C);
J:=inter_unique(segment(N,C),s);J1:=symetrie(N,J);
P1:=polygone(M,I,A):;Q1:=polygone(M,I1,B):;
P2:=polygone(M,B,J,N):;
P3:=polygone(N,J1,I):;Q3:=polygone(N,J,I2):;
P4:=polygone(I,J1,A):;Q4:=polygone(I2,J,C):;
P5:=polygone(J,B,C):;
affichage(P1,1+rempli);affichage(Q1,1+rempli);
affichage(P2,2+rempli);
affichage(P3,3+rempli);affichage(Q3,3+rempli);
affichage(P4,4+rempli);affichage(Q4,4+rempli);
affichage(P5,5+rempli);
affichage(translation(14,Q1),1+rempli);
affichage(translation(14,P2),2+rempli);
affichage(translation(14,Q3),3+rempli);
affichage(translation(8+6*i,Q4),4+rempli);
affichage(translation(8+6*i,P5),5+rempli);
```

On obtient un puzzle de 5 pièces :



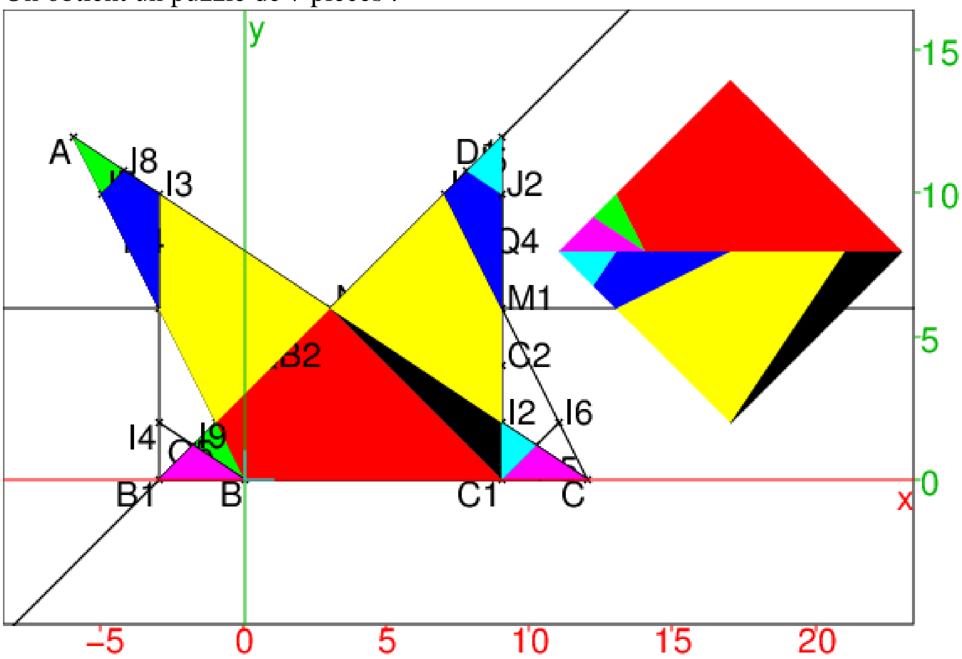
6.5 On transforme le triangle récalcitrant en un triangle rectangle isocèle

Premier essai : On pose le triangle rectangle isocèle C_1, B_1, A_1 sur le triangle récalcitrant pour que A_1B_1 passe par N.
On tape :

```
A:=point(-6+12*i);B:=point(0);C:=point(12);
D1:=point(9+12*i);B1:=point(-3);C1:=point(9);
triangle(A,B,C);triangle(D1,B1,C1);
I:=inter_unique(droite(B1,D1),droite(A,B));
I1:=point(6*i);
I2:=inter_unique(droite(A,C),droite(x=9));
M:=milieu(A,B);N:=milieu(A,C);
P0:=polygone(N,C1,I2):; P1:=polygone(N,I,B,C1):;
affichage(P0,rempli); affichage(P1,1+rempli);
C2:=symetrie(I2,C1);B2:=symetrie(I,B1);
I3:=symetrie(N,I2);
M1:=symetrie(N,M);I0:=symetrie(N,I);;
polygone(N,M1,C);
I4:=point(-3,2);I5:=point(12-9/5,6/5);
P2:=polygone(A,I7,I8):;
Q2:=polygone(B,I,symetrie(M,I8)):;
P3:=polygone(N,I3,M,I):;Q3:=polygone(N,I2,M1,I0):;
affichage(P2,2+rempli);affichage(Q2,2+rempli);
affichage(P3,3+rempli);affichage(Q3,3+rempli);
I6:=symetrie(M1,I0);segment(C1,I6);
```

```
I7:=symetrie(N,I6);;segment(B,I4);
s:=segment(I7,-3+12*i):;
I8:=inter_unique(s,droite(A,C));
P4:=polygone(M,I3,I8,I7);Q4:=polygone(M1,I0,J5,J2);
affichage(P4,4+rempli);affichage(Q4,4+rempli);
I9:=symetrie(M,I8);
P5:=polygone(C1,I5,C);Q5:=polygone(B1,B,I9);
affichage(P5,5+rempli);affichage(Q5,5+rempli);
J5:=symetrie(M1,I5);J2:=symetrie(M1,I2);
P6:=polygone(C1,I5,I2):;Q6:=polygone(D1,J5,J2):;
affichage(P6,6+rempli);affichage(Q6,6+rempli);
T0:=translation(14+8*i,rotation(C1,pi/2,P0));
T3:=translation(14+8*i,rotation(C1,pi/2,Q3)):;
T4:=translation(14+8*i,rotation(C1,pi/2,Q4)):;
T6:=translation(14+8*i,rotation(C1,pi/2,Q6)):;
T1:=translation(14+8*i,P1):;
T2:=translation(14+8*i,Q2):;
T5:=translation(14+8*i,Q5):;
affichage(T0,rempli); affichage(T1,1+rempli);
affichage(T2,2+rempli); affichage(T3,3+rempli);
affichage(T4,4+rempli); affichage(T5,5+rempli);
affichage(T6,6+rempli);
```

On obtient un puzzle de 7 pièces :

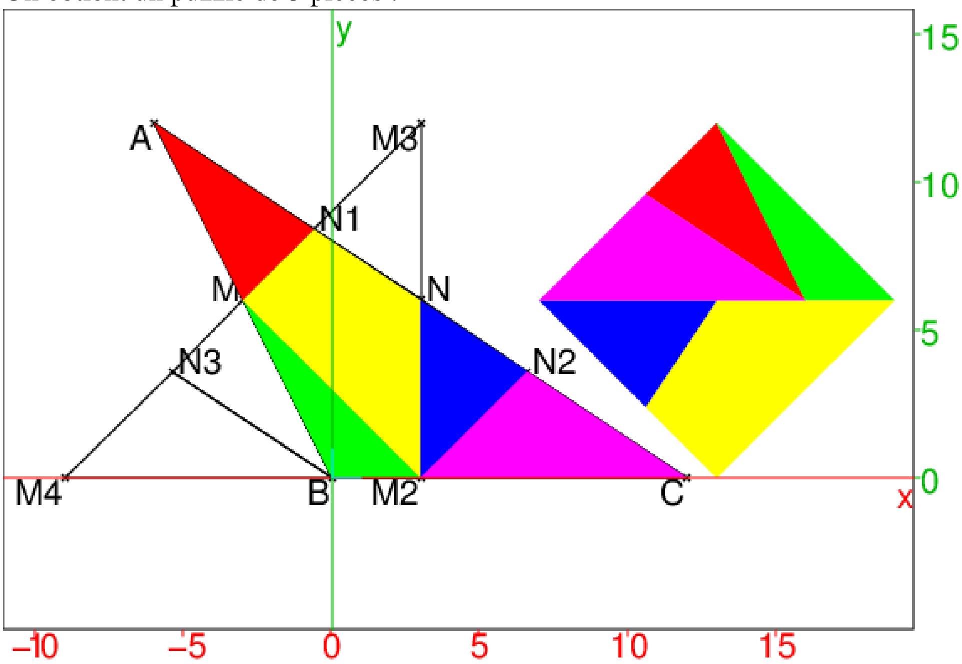


Deuxième essai : On pose le triangle rectangle isocèle C_1, B_1, A_1 sur le triangle récalcitrant pour que A_1C_1 passe par N.
On tape :

```
A:=point(-6+12*i);B:=point(0);C:=point(12);
triangle(A,B,C);
M:=affichage(milieu(A,B),quadrant2);
```

```
N:=milieu(A,C);M2:=point(3);
M3:=point(3+12*i);M4:=point(-9);
N1:=inter_unique(droite(A,C),droite(M3,M4));
N2:=inter_unique(droite(A,C),droite(y=x-3));
N3:=symetrie(M,N1);
P1:=polygone(A,M,N1):;
P2:=polygone(M,B,M2):;
P3:=polygone(N1,M,M2,N):;
P4:=polygone(N,M2,N2):;
P5:=polygone(M2,C,N2):;
Q1:=polygone(B,M,N3):;
Q4:=symetrie(N,P4):;
Q5:=polygone(M4,B,N3):;
affichage(P1,1+rempli);affichage(P2,2+rempli);
affichage(P3,3+rempli);affichage(P4,4+rempli);
affichage(P5,5+rempli);
affichage(translation(16+6*i,Q1),1+rempli);
affichage(translation(16+6*i,Q2),2+rempli);
affichage(translation(16+6*i,rotation(M2,pi/2,Q3)),3+rempli);
affichage(translation(16+6*i,rotation(M2,pi/2,Q4)),4+rempli);
affichage(translation(16+6*i,Q5),5+rempli);
```

On obtient un puzzle de 5 pièces :



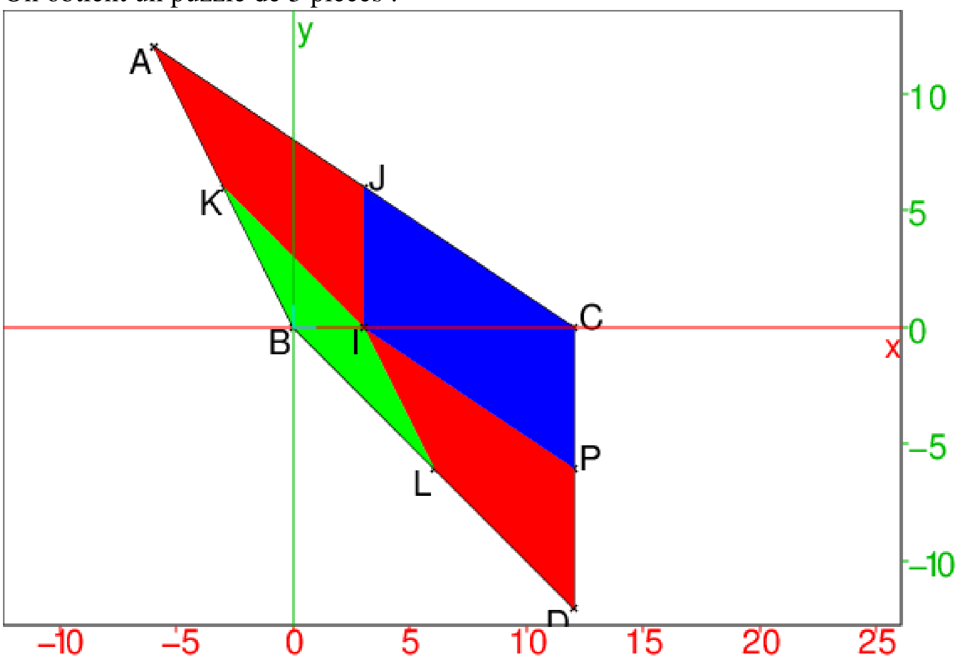
6.6 Décomposer des triangles de même hauteur et de même base en 3 pièces

En général, on peut décomposer des triangles de même hauteur et de même base en 3 pièces.
On considère par exemple le triangle récalcitrant ABC et le triangle rectangle isocèle CBD où D est le point d'affixe $12 - 12 * i$.

Soient I l'intersection de AD avec BC et les 2 parallélogrammes :
$BLIK$ (IK (resp IL) est parallèle à BD (resp BA) et
$CJIR$ (IJ (resp IP) est parallèle à CD (resp CA).
Le polygone A, K, I, J se déduit du polygone I, L, D, P par translation de vecteure $A - C$ et les autres pièces sont symétriques par rapport à BC On tape :

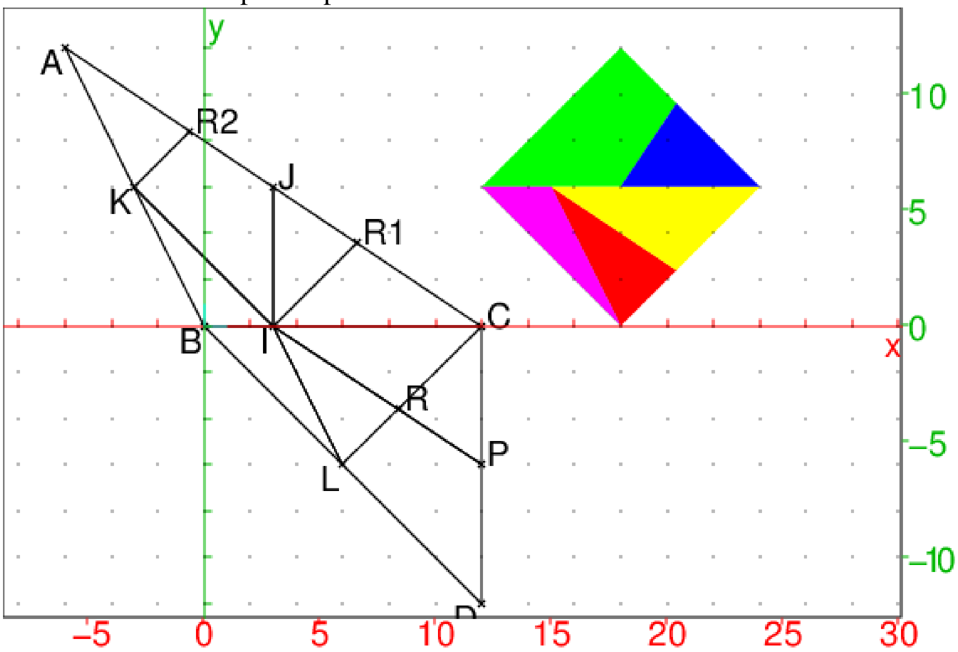
```
A:=point(-6+12*i);B:=point(0);
C:=affichage(point(12),quadrant1);
D:=point(12-12*i);I:=point(3);
triangle(A,B,C);triangle(D,B,C);
J:=affichage(point(3+6*i),quadrant1);
J:=point(3+6*i);K:=point(-3+6*i);
P:=affichage(point(12-6*i),quadrant1);
P:=point(12-6*i);
P1:=polygone(A,K,I,J);
Q1:=polygone(I,L,D,P);
P2:=polygone(K,B,I);
Q2:=polygone(L,B,I);
P4:=polygone(J,I,C);
Q4:=polygone(P,I,C);
affichage(P1,1+rempli);
affichage(Q1,1+rempli);
affichage(P2,2+rempli);
affichage(Q2,2+rempli);
affichage(P4,4+rempli);
affichage(Q4,4+rempli);
I:=I;
```

On obtient un puzzle de 3 pièces :



Pour transformer le triangle récalcitrant ABC en un carré il suffit de tracer le segment CL qui va partager $Q1$ en rouge et jaune et $Q3$ en bleu et cyan ce qui donne

en tout 5 pièces pour tansformer ABC en un carré.
On retrouve ainsi le puzzle précédent :



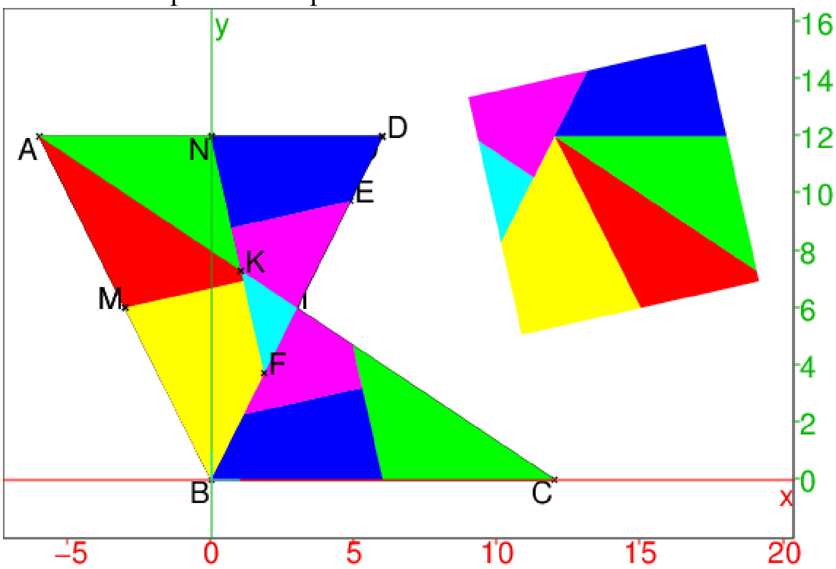
6.7 On transforme le triangle en un triangle isocèle

On tape :

```
A:=point(-6+12*i);B:=point(0);C:=point(12);
D:=point(6+12*i);triangle(A,B,C);triangle(A,B,D);
M:=affichage(milieu(A,B),quadrant2);N:=point(12*i);
F:=inter_unique(segment(0,D),cercle(N,,6*sqrt(2)));
E:=translation((D-B)/2,F);E1:=projection(droite(N,F),E);
M1:=projection(droite(N,F),M);segment(N,F);I:=milieu(A,C);
K:=inter_unique(segment(N,F),droite(A,C));
P1:=polygone(A,M,M1,K):;Q1:=P1:;
Q2:=polygone(A,K,N):;P2:=symetrie(I,Q2):;
P3:=polygone(B,M,M1,F):;Q3:=P3:;
Q4:=polygone(D,N,E1,E):;P4:=symetrie(I,Q4):;
Q5:=polygone(I,K,E1,E):;P5:=symetrie(I,Q5):;
P6:=polygone(I,K,F):;Q6:=P6
Q3:=symetrie(M,P3):;Q4:=symetrie(I,P4):;
T5:=translation(2*(N-E)+18,P5):;
affichage(translation(18,Q1),1+rempli);
affichage(translation(18,Q2),2+rempli);
affichage(translation(18,symetrie(M,Q3)),3+rempli);
affichage(translation((point(12+12*i)-B),P4),4+rempli);
affichage(translation(2*(N-E)+18,Q5),5+rempli);
affichage(translation(2*(N-E)+18,Q6),6+rempli);
affichage(P1,1+rempli);affichage(P2,2+rempli);
affichage(P3,3+rempli);affichage(P4,4+rempli);
affichage(P5,5+rempli);affichage(P6,6+rempli);
```

```
affichage(Q1,1+rempli);affichage(Q2,2+rempli);
affichage(Q3,3+rempli);affichage(Q4,4+rempli);
affichage(Q5,5+rempli);affichage(Q6,6+rempli);
D:=affichage(D,quadrant1);M:=M;N:=N;K:=K;F:=F;
```

On obtient un puzzle de 6 pièces :



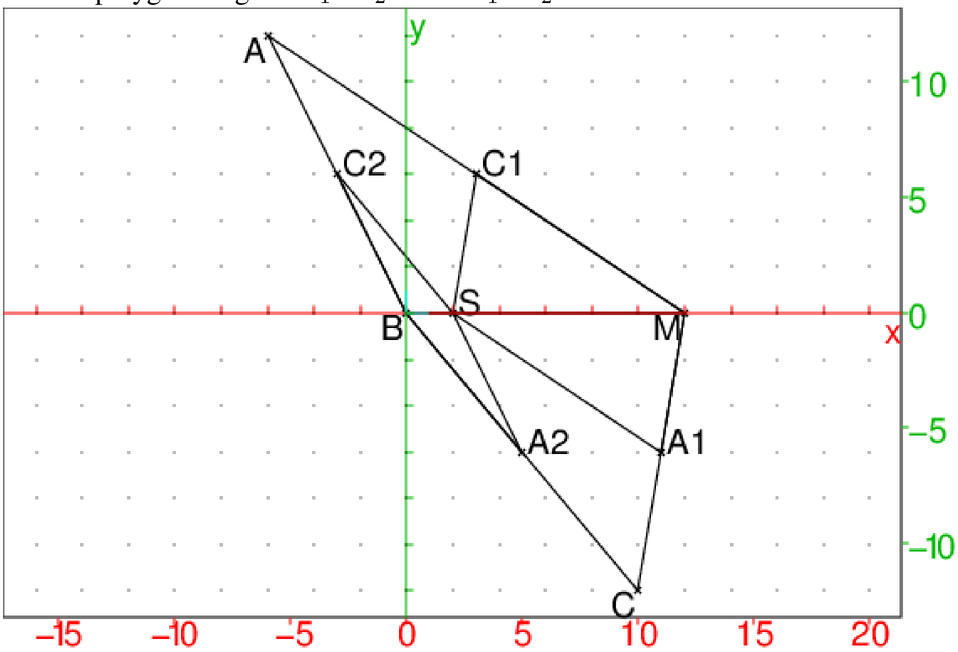
7 Décomposer des polygones de même aire

Soient 2 polygones P_1 et P_2 ayant la même aire.
Peut-on toujours partager le polygone $P-1$ pour que les pièces de ce puzzle reconstitue le polygone P_2 ?
La réponse est oui et cela a été démontré par F.Bolyai (1832) et de façon undépendante par P Gerwien.
La démonstration provient essentiellement des faits suivants :

1. Deux triangles qui ont la même base et la même hateur peuvent être décomposer par un puzzle de 3 pièces.
   On tape :

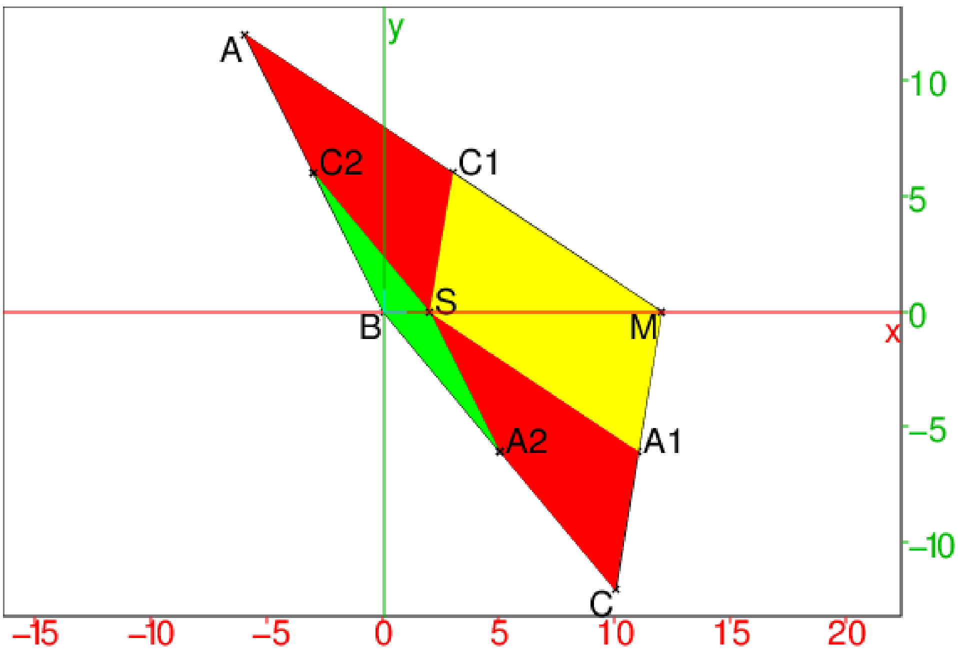
```
A:=point(-6+12*i);B:=point(0);
M:=point(12);C:=point(10-12*i)
triangle(B,M,C);triangle(B,M,A);
S:=inter_unique(droite(A,C),droite(B,M));
d1:=parallele(S,droite(M,C)):;
d2:=parallele(S,droite(B,C)):;
C1:=inter_unique(d1,droite(A,M));
C2:=inter_unique(d2,droite(A,B));
A1:=inter_unique(parallele(S,droite(M,A)),droite(C,M));
A2:=inter_unique(parallele(S,droite(B,A)),droite(C,B));
P2:=polygone(S,C2,B);
Q2:=polygone(S,A2,B);
P3:=polygone(S,C1,M);
Q3:=polygone(A1,S,M);
```

On obtient 2 parallélogrammes (les polygones $A2, S, C2, B$ et $C1, S, A1, M$) et deux polygones égaux C_1AC_2S et CA_1SA_2 :



Pour avoir de la couleur on rajoute :

```
affichage(polygone(C1,A,C2,S),1+rempli);
affichage(polygone(A1,S,A2,C),1+rempli);
affichage(P2,2+rempli);
affichage(Q2,2+rempli);
affichage(P3,3+rempli);
affichage(Q3,3+rempli);
```

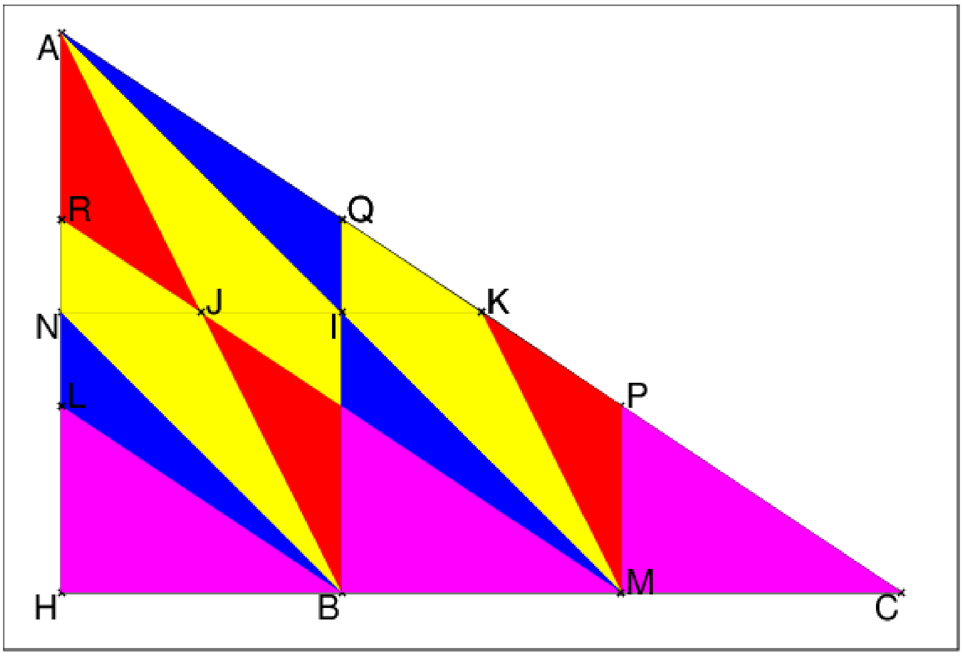


2. n triangles de même aire. Si ils ont même sommet A et si les bases B_0B_1,

B_1B_2, B_2B_3 ...$B_{n-1}B_n$ sont telles que : $\overrightarrow{B_0B_1} = \overrightarrow{B_1B_2} = ...\overrightarrow{B_{n-1}B_n}$
alors ils sont chacun décomposables avec un puzzle de $n+1$ pièces.
Pour réaliser ce puzzle il suffit de tracer à partir des sommets :
$B_1...B_{n-1}$ les parallèles aux côtés : $AB_0, AB_1, ...AB_{n-1}, AB_n$
Par exemple, on partage le triangle récalcitrant en 2 triangles de même aire
et on l'accole au triangle rectangle AHB.

```
A:=point(-6+12*i);B:=point(0);C:=point(124;
triangle(A,B,C);
M:=milieu(B,C);H:=point(-6);I:=point(6*i);
segment(L,B);segment(N,B);segment(M,R);segment(N,K); ;
N:=point(-6+6*i);
J:=inter_unique(droite(A,B),parallele(M,droite(A,C)));
K:=inter_unique(droite(A,C),parallele(M,droite(A,B)));
L:=inter_unique(droite(A,H),parallele(B,droite(J,M)));
O:=inter_unique(droite(J,M),droite(x=0));
P:=inter_unique(droite(A,C),droite(x=6));
Q:=inter_unique(droite(A,K),droite(x=0));
R:=inter_unique(droite(O,J),droite(x=-6));
T1:=polygone(A,R,J):; T2:=polygone(N,R,J):;
T3:=polygone(B,J,N):; T4:=polygone(B,L,N):;
T5:=polygone(B,H,L):;
P1:=polygone(B,J,O):;P2:=polygone(I,O,J):;
P3:=polygone(A,J,I):;P4:=polygone(I,M,O):;
P5:=polygone(B,M,O):;
Q1:=polygone(M,K,P):; Q2:=polygone(I,Q,K):;
Q3:=polygone(M,K,I):;Q4:=polygone(I,A,Q):;
Q5:=polygone(C,M,P):;Q5;;
affichage(T1,1+rempli);affichage(P1,1+rempli);
affichage(Q1,1+rempli);
affichage(T2,3+rempli);affichage(P2,3+rempli);
affichage(Q2,3+rempli);
affichage(T3,3+rempli);affichage(P3,3+rempli);
affichage(Q3,3+rempli);
affichage(T4,4+rempli);affichage(P4,4+rempli);
affichage(Q4,4+rempli);;
affichage(T5,5+rempli);affichage(P5,5+rempli);
affichage(Q5,5+rempli);;
```

On obtient :



8 Puzzle transformant le triangle récalcitrant en carré

On change dans ce qui précède

```
affichage(T2,3+rempli);
affichage(P2,3+rempli);
affichage(Q2,3+rempli);
```

en le remplaçant par

```
affichage(T2,2+rempli);
affichage(P2,2+rempli);
affichage(Q2,2+rempli);
```

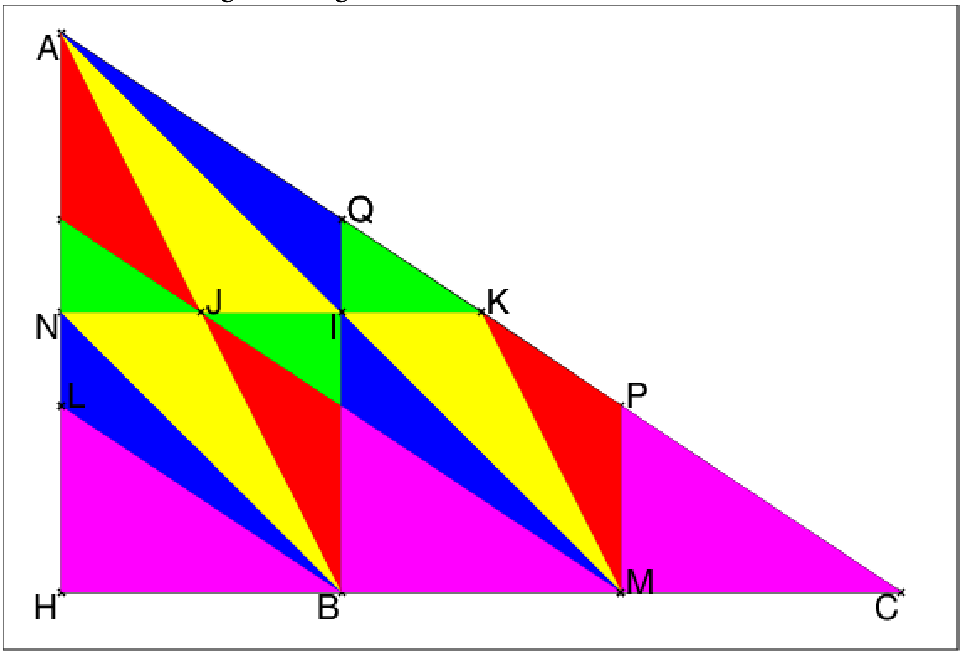
On tape donc :

```
puzzle10():={
local S,A,B,C,M,N,H,I,J,K,L,O,P,Q,R,T1,P1,Q1,T2,P2,Q2,
      T3,P3,Q3,T4,P4,Q4,T5,P5,Q5;
S:=NULL;
A:=point(-6+12*i);B:=point(0);C:=point(12);
triangle(A,B,C);
M:=milieu(B,C);H:=point(-6);I:=point(6*i);
segment(L,B);segment(N,B);segment(M,R);segment(N,K); ;
N:=point(-6+6*i);
J:=inter_unique(droite(A,B),parallele(M,droite(A,C)));
K:=inter_unique(droite(A,C),parallele(M,droite(A,B)));
L:=inter_unique(droite(A,H),parallele(B,droite(J,M)));
O:=inter_unique(droite(J,M),droite(B,I));
```

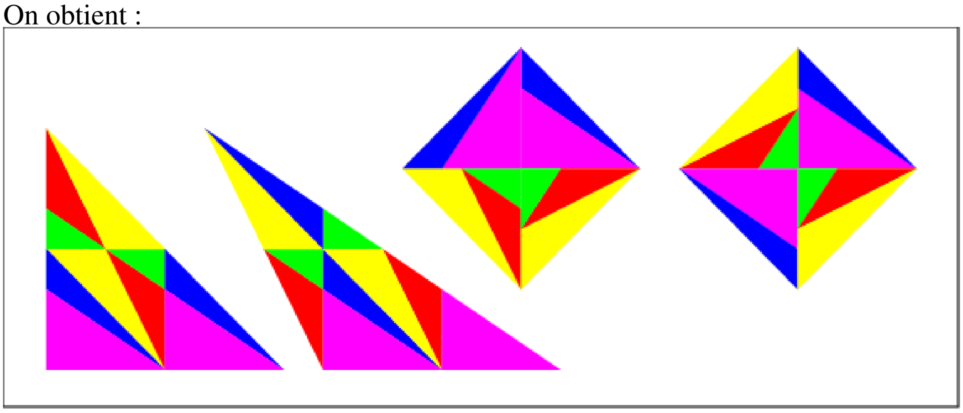
```
P:=inter_unique(droite(A,C),droite(x=6));
Q:=inter_unique(droite(A,C),droite(B,I));
R:=inter_unique(droite(O,J),droite(A,N));
T1:=polygone(A,R,J):;T2:=polygone(N,R,J):;
T3:=polygone(B,J,N):;T4:=polygone(B,L,N):;
T5:=polygone(B,H,L):;
P1:=polygone(B,J,O):;P2:=polygone(O,I,J):;
P3:=polygone(A,J,I):;P4:=polygone(O,M,I):;
P5:=polygone(B,M,O):;
Q1:=polygone(M,K,P):;Q2:=polygone(I,Q,K):;
Q3:=polygone(M,K,I):;Q4:=polygone(I,A,Q):;
Q5:=polygone(C,M,P):;
S:=S,affichage(P1,1+rempli);
S:=S,affichage(P2,2+rempli);
S:=S,affichage(P3,3+rempli);
S:=S,affichage(P4,5+rempli);
S:=S,affichage(P5,5+rempli);
S:=S,affichage(Q1,1+rempli);
S:=S,affichage(Q2,2+rempli);
S:=S,affichage(Q3,3+rempli);
S:=S,affichage(Q4,4+rempli);
S:=S,affichage(Q5,5+rempli);
S:=S,affichage(translation(14+10*i,T4),4+rempli);
S:=S,affichage(translation(14+10*i,T5),5+rempli);
S:=S,affichage(translation(8+4*i,T3),3+rempli);
S:=S,affichage(translation(8+4*i,P2),2+rempli);
S:=S,affichage(translation(8+4*i,P1),1+rempli);
S:=S,affichage(translation(14+10*i,rotation(H,pi/2,T5)),5+rempli);
S:=S,affichage(translation(14+10*i,rotation(H,pi/2,T4)),4+rempli);
S:=S,affichage(translation(20+4*i,rotation(H,pi/2,T3)),3+rempli);
S:=S,affichage(translation(8+4*i,rotation(I,pi/2,P1)),1+rempli);
S:=S,affichage(translation(8+4*i,rotation(I,pi/2,P2)),2+rempli);
S:=S,affichage(translation(30+10*i,T5),5+rempli);
S:=S,affichage(translation(30+10*i,symetrie(H,T5)),5+rempli);
S:=S,affichage(translation(30+10*i,T4),4+rempli);
S:=S,affichage(translation(30+10*i,symetrie(H,T4)),4+rempli);
S:=S,affichage(translation(24+4*i,rotation(I,pi/2,P2)),2+rempli);
S:=S,affichage(translation(24+4*i,rotation(I,pi/2,P1)),1+rempli);
S:=S,affichage(translation(24+4*i,rotation(I,pi/2,T3)),3+rempli);
S:=S,affichage(translation(24+4*i,rotation(I,-pi/2,P2)),2+rempli);
S:=S,affichage(translation(24+4*i,rotation(I,-pi/2,P1)),1+rempli);
S:=S,affichage(translation(24+4*i,rotation(I,-pi/2,T3)),3+rempli);
S:=S,affichage(translation(-8,T1),1+rempli);
S:=S,affichage(translation(-8,T2),2+rempli);
S:=S,affichage(translation(-8,T3),3+rempli);
S:=S,affichage(translation(-8,T4),4+rempli);
S:=S,affichage(translation(-8,T5),5+rempli);
S:=S,affichage(translation(-8,P1),1+rempli);
```

```
S:=S,affichage(translation(-8,P2),2+rempli);
S:=S,affichage(translation(-8,P3),3+rempli);
S:=S,affichage(translation(-8,P4),4+rempli);
S:=S,affichage(translation(-8,P5),5+rempli);
return S;
}:;
```

On obtient un puzzle de de 2 fois 5 pièces transformant le triangle récalcitrant en carré et en un triangle rectangle isocèle :



On remarque que chacun des triangles AHB,ABM et AMC se décompose selon le carré $NHBI$ qui est formé de 5 pièces et qui de décompose facilement en 2 triangles rectangles isocèles.
On tape :
`puzzle10()`
On obtient :



On peut maintenant rassembler certaines pièces (à vous de les trouver) pour avoir un puzzle de 7 pièces pour ABC.