

Examen de la session de juin
2 heures, documents non autorisés

Rédiger vos réponses comme demandé sur le sujet et glissez-le dans la copie réglementaire sur laquelle vous aurez également indiqué vos noms, prénoms et numéro de carte d'étudiant. Sur cette copie, vous rédigerez aussi les réponses de l'exercice 5.

Rappel : Quand une commande est suivie du délimiteur \$, Maxima n'affiche pas ce qu'il a effectué dans son noyau de calcul relativement à cette commande. Respectez cette règle quand vous rencontrerez ci-dessous ces délimiteurs dollar.

Exercice 1. Ecrire dans les zones blanches, ce que donne Maxima quand on exécute la cellule suivante :

```
(%i1) L: [b,6,-4,3]$ L[3];  
(%o1)  
  
(%i2) length(L)+L[1];  
(%o2)  
  
(%i3) L[length(L)]:0$ L;  
(%o3)  
  
(%i4) L:append(L, [Bond,0,0,7])$ L;  
(%o4)
```

Exercice 2. Ecrire dans la zone blanche, ce que donne Maxima quand on exécute la cellule suivante:

```
(%i5)f(x):=(x+1)/(x+2)$
if f(3/2)>1 then (
  M:makelist(2*j+3,j,0,4),
  S:0,
  for k:1 thru length(M) do(
    S:S+M[k]
  )
)
else (
  M:makelist(2*j-2,j,2,5),
  S:0,
  for k:1 thru length(M) do(
    S:S+M[k]
  )
)$
M;S;
(%o5)
```

Exercice 3. Ecrire dans les zones blanches, ce que donne Maxima quand on exécute la cellule suivante :

```
(%i6)  trigrat(cos(x)^2);
(%o6)

(%i7)  trigexpand(sin(a-%pi/4));
(%o7)
```

Exercice 4. Le crible d'Erathostène dans sa "version papier", consiste à suivre les démarches suivantes :

1. Choisir un entier M .
2. Placer tous les entiers de 1 à M dans un tableau T et ceci sur votre feuille de papier. On considère que 1 n'est pas premier, barrez-le.
3. Pour chaque entier k entre 2 et \sqrt{M} , on regarde si k est encore présent dans le tableau T . Si c'est le cas, alors k est un nombre premier et on barre tous ses multiples $k \cdot u$ avec u entre 2 et M/k .
4. A la fin de cette double boucle, le tableau T contient exactement les nombres premiers entre 2 et M .

Dans son implémentation avec **Maxima**, nous choisissons ici de considérer une liste **T** qui ne contient au départ que des 1. Si l'on veut déterminer tous les premiers jusque $M = 100$, on commencera donc par :

```
(%i8) M:100;  
      T:makelist(1,k,1,M)$  
      T[1]:0$
```

On a déjà indiqué ci-dessus que 1 n'est pas premier.

Barrer un entier i du tableau sera effectué dans votre programme en affectant 0 à la i -ème place de la liste **T**. Par exemple, pour barrer 6, c'est à dire 2 fois 3, il vous faudra affecter dans votre programme `T[2*3]:0`.

1. Programmer ci-dessous le crible d'Erathostène à l'aide d'une double boucle. A la fin de cette première phase, vous disposerez de la liste **T** telle que, pour tout i , $T[i]$ égal à 1 est équivalent au fait que i est un nombre premier.
2. Créer à la suite et en commandes **Maxima** une liste vide nommée **prem**. Puis, à l'aide d'une boucle, y ajouter itérativement les nombres premiers (quand $T[i]$ vaut 1, l'entier i est premier et on l'ajoute à la liste **prem** avec la commande `endcons`).

Exercice 5. Rappelons que si R est un réel strictement positif, l'ensemble des points du plan de coordonnées (x, y) vérifiant

$$(x - x_A)^2 + (y - y_A)^2 - R^2 = 0$$

est le cercle de centre $A(x_A, y_A)$ et de rayon R . Rappelons aussi que l'ensemble des points de coordonnées (x, y) vérifiant l'équation

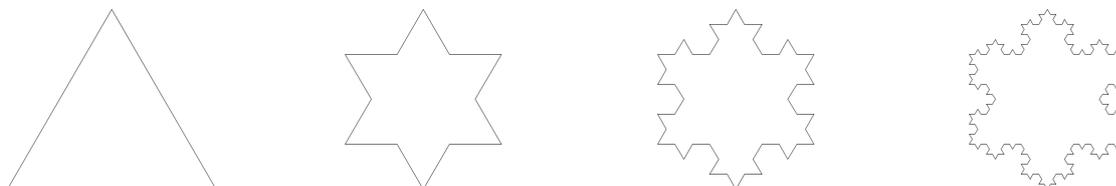
$$(x_B - x_A) \times (y - y_A) - (x - x_A) \times (y_B - y_A) = 0$$

est l'équation de la droite passant par $A(x_A, y_A)$ et $B(x_B, y_B)$.

1. Reproduire sur votre copie, le dessin obtenu avec Maxima avec le programme ci-dessous.
2. Quel est le but de cette construction?

```
(%i9)norme(A,B):=sqrt((A[1]-B[1])^2+(A[2]-B[2])^2)$
polycercle(A,B):=(x-A[1])^2+(y-A[2])^2-norme(A,B)^2$
polydroite(A,B):=(B[1]-A[1])*(y-A[2])-(x-A[1])*(B[2]-A[2])$
cercle(A,B):=block([R],R:norme(A,B),
                    parametric(A[1]+R*cos(t),A[2]+R*sin(t),t,0,2*%pi))$
A:[0,0]$B:[2,1]$C:[3,4]$
s1:solve([polycercle(A,B)=0,polydroite(A,C)=0],[x,y])$
B1:subst(s1[2],[x,y])$ /*on admet ici que les coordonnées de B1
                        sont positives*/
s2:solve([polycercle(B1,A)=0,polycercle(B,A)=0],[x,y])$
M1:subst(s2[1],[x,y])$ /*on admet ici que le point M1 n'est pas A*/
e:0.25$
wxdraw2d(
proportional_axes='xy,point_type=circle,color=black,
points([A,B,B1,C,M1]),
points_joined=true,color=red,
points([A,B]),points([A,C]),points([A,M1]),
color=black,nticks=100,color=black,cercle(A,B1),
color=blue,cercle(B,A),cercle(B1,A),
xrange=[-2.5,4.5],yrange=[-2.5,4.5],
color=black,label(["A",A[1]-e,A[2]-e]),label(["B",B[1]+e,B[2]]),
label(["B1",B1[1]-e,B1[2]]),label(["C",C[1]+e,C[2]]),
label(["M1",M1[1]+e,M1[2]+e])
);
```

Exercice 6. Voici les différentes étapes de la construction du flocon de von Koch par application successive de la même transformation.



Le triangle correspond au flocon de von Koch avec un degré de détail valant 1 et la dernière figure est le flocon de Koch de degré 4. On note F_d le flocon de degré d . Pour passer du flocon F_d au flocon F_{d+1} on applique la transformation suivante à tous les segments de F_d .



Le but de cet exercice est créer une procédure **Maxima** retournant le flocon de von Koch F_d pour un degré de précision d quelconque. Soit (O, \vec{i}, \vec{j}) un repère orthonormée du plan.

1. Indiquer la réponse de **Maxima** lors de l'exécution de la cellule suivante :

```
(%i10) point(z):=block([x,y],
      x:realpart(z),
      y:imagpart(z),
      return([x,y])
    )$
(%i11) point(3+5%i);
(%o11) [3,5]
```

2. Compléter la fonction suivante retournant l'affixe d'un point P (un exemple d'exécution est donné).

```
(%i12) affixe(P):=block([x,      ],
      x:P[1],

    )$
(%i13) affixe([2,-1]);
(%o13) 2-%i
```

3. Soient trois points A, B et C d'affixes respectifs z_A, z_B et z_C .

a. Quelle est l'affixe du translaté de A par le vecteur \overrightarrow{BC} ?

b. Décrire géométriquement le point dont l'affixe est $e^{i\pi/3}z_A$

c. Même question pour $z_B + e^{i\pi/3}(z_A - z_B)$

4. Compléter la fonction `elemKoch` suivante qui étant donnée deux points A et B retourne la liste des cinq points A, C, D, E, B correspondant à la transformation élémentaire. On utilisera naturellement les résultats de l'exercice précédent et les fonctions `affixe` et `point`.

```
(%io14) elemKoch(A,B):=block([zA, ],
    zA:affixe(A)

    return([A,C,D,E,B])
)$
```