Correction du devoir maison:

Exercice 1:
1)

```
> restart;
  f:=x->sqrt(x+1)/(x-3);
```

$$f := x \to \frac{\sqrt{x+1}}{x-3} \tag{1}$$

2) L'ensemble de définition de f est [-1,3[u]3,infinity[.
3)

```
> iscont(f(x),x=-1..3);
```

$$true \tag{2}$$

Attention par défaut, la commande **iscont** teste la continuité sur un intervalle ouvert. Ici on n'a donc pas testé si f était continue en -1, on écrit donc:

```
> iscont(f(x),x=-1..2,'closed'); # pour tester sur [-1,2]
  iscont(f(x),x=2..3); # pour tester sur ]2,3[
```

$$true \tag{3}$$

$$true$$

et enfin:

```
> iscont(f(x),x=3..infinity);
```

$$true \tag{4}$$

4) Il faut tester la limite en -1, en 3 et en l'infini. Attention en 3, il faut préciser si l'on veut les limites à gauche ou à droite.

```
> l_moins_1:=limit(f(x),x=-1);
  l_infini:=limit(f(x),x=infinity);
  l_trois_gauche:=limit(f(x),x=3,left);
  l_trois_droite:=limit(f(x),x=3,right);
```

$$l\_moins\_1 := 0$$

$$l\_infini := 0$$

$$l\_trois\_gauche := -\infty$$

$$l\_trois\_droite := \infty \tag{5}$$

5)

```
> g1:=D(f); # ou, autre solution:
  g2:=x->diff(f(x),x);
  g2(x);
```

$$g1 := x \to \frac{1}{2}\,\frac{1}{\sqrt{x+1}\,(x-3)} - \frac{\sqrt{x+1}}{(x-3)^2} \tag{6}$$

$$g2 := x \to \frac{d}{dx} f(x)$$

$$\frac{1}{2}\,\frac{1}{\sqrt{x+1}\,(x-3)} - \frac{\sqrt{x+1}}{(x-3)^2}$$

6)

```
> solve(g1(x)<0);
```

$$RealRange(Open(-1), Open(3)), RealRange(Open(3), \infty) \tag{7}$$

Exercice 2:
A partir de la liste de départ, on construit une liste qui nous permettra de remplir la matrice.
Par ex si L:=[1,2,3]; on va construire la liste liste_matrice qui va être égale à [1,2,3,3,1,2,2,3,1]

```
> restart;
> circulante:=proc(L)
    local long,i,liste_matrice;
    long:=nops(L): # nombre de termes de la liste L
    liste_matrice:=L;
    for i from 1 to long-1 do
     liste_matrice:=[op(liste_matrice),op(L[long-i+1..long]),op(L[1..
  long-i])];
    od:
  RETURN(linalg[matrix](long,long,liste_matrice))
  end:
> circulante([1,2,3]);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{bmatrix} \tag{8}$$

Exercice 3:

1) 1/ Je ne refais pas le dessin exigé, si vous n'avez pas réussi posez moi la question.

2/ La méthode récursive consiste à déterminer si la racine est entre a et c (respectivement entre c et b) et ensuite à rappeler la procédure dichotomie aux bonnes variables, ce qui donne:

```
> dichotomie:=proc(f,a,b,n)
    local c;
    c:=(a+b)*0.5;
```

```
    if n>1 then
    if f(a)*f(c)<0 then  # on teste si la racine est entre a et c
    dichotomie(f,a,c,n-1);  # dans ce cas on réapplique la procédure
   entre a et c, n-1 fois
    else dichotomie(f,c,b,n-1); # sinon on la réapplique entre c et
   b.
    fi;
    else c;
    fi;
    end:
```

3/ On teste la procédure:

```
> f:=x->x**2-2:
  a:=1.0:
  b:=2.0:
  n:=20:
> dichotomie(f,a,b,n);
```

$$1.414214134 \tag{9}$$

On obtient une approximation de racine de 2. Pour connaitre la précision, on demande à Maple:

```
> evalf(sqrt(2)-dichotomie(f,a,b,n));
```

$$-5.72\ 10^{-7} \tag{10}$$

On a une précision à 10^(-6) près.

Variante:
Si on ne veut pas faire de manière récursive, on a envie d'écrire:

```
> dichotomie_erreur:=proc(f,a,b,n)
    local c;
    while n>0 do
    c:=(a+b)*0.5;
    n:=n-1;
    if f(a)*f(c)<0 then
    b:=c;
    else a:=c;
    fi;
    od;
    RETURN(c)
    end:
```

Maple ne dit rien à ce moment là, mais si on essaie de l'appliquer à la fonction f, on obtient:

```
> dichotomie_erreur(f,a,b,n);
Error, (in dichotomie_erreur) illegal use of a formal parameter
```

Ceci vient du fait que l'on modifie la valeur des paramètres. Par exemple on écrit n:= n-1; ou b:=c; et Maple n'autorise pas cela. Pour contourner ce problème, on crée des variables locales que l'on pourra elles modifier.

```
> dichotomie_bis:=proc(f,a,b,n)
    local c,n_loc,a_loc,b_loc;
    n_loc:=n; # on initiale nos nouvelles variables
    a_loc:=a; # et maintenant on peut refaire exactement pareil
    b_loc:=b; # mais en utilisant les variables _loc!
    while n_loc>0 do
    c:=(a_loc+b_loc)*0.5;
    n_loc:=n_loc-1;
    if f(a_loc)*f(c)<0 then
    b_loc:=c;
    else a_loc:=c;
    fi;
    od;
    RETURN(c)
    end:
> dichotomie_bis(f,a,b,n);
```

$$1.414214134 \tag{11}$$

2) 1/ D'après le théorème de Thalès et le dessin qu'il fallait faire, le point c représente l'intersection de la droite (f(a) f(b)) et de l'axe des abcisses.
2/ Je ne fais pas le dessin demandé.
3/ Il n'y a presque rien à modifier, uniquement à adapter la facon dont on calcule c.
Attention: il y avait une erreur dans l'énoncé (merci aux étudiants qui me l'ont fait remarquer), on obtenait:
c=a-(a-b)*f(a)/(f(a)-f(b))

```
> regula:=proc(f,a,b,n)
    local c;
    c:=a - (a-b)*f(a)/(f(a)-f(b));
    if n>1 then
    if f(a)*f(c)<0 then  # on teste si la racine est entre a et c
    regula(f,a,c,n-1);  # dans ce cas on réapplique la procédure
   entre a et c, n-1 fois
    else regula(f,c,b,n-1); # sinon on la réapplique entre c et b.
    fi;
    else c;
    fi;
    end:
> regula(f,a,b,n);
```

$$1.414213562 \tag{12}$$

```
> evalf(regula(f,a,b,n)-sqrt(2));
```

$$0. \tag{13}$$

evalf utilisé sans option donne une approximation avec 10 chiffres signicatifs, ici ce n'est pas suffisant. On tatonne un peu et on lui demande suffisamment de chiffres après la virgule:

```
> evalf(regula(f,a,b,n)-sqrt(2),20);
```

$$-2.371\,10^{-16} \tag{14}$$

On obtient une précision à 10^(-15) près. En seulement 20 itérations, on obtient trois fois plus de chiffres significatifs.

3) 1/ Le point c représente l'intersection de l'axe des abcisses avec la tangente de f au point a.

3/ La procédure est plus simple que précédemment étant donné qu'il n'y a pas de test à effectuer pour savoir où est la racine par rapport à a, b et c.

```
> newton:=proc(f,a,n)
   local c;
   c:=a-f(a)/D(f)(a);
   if n>1
    then newton(f,c,n-1);
    else
    c;
  fi;
  end:
```

```
> newton(f,a,n);
```

$$1.414213562 \tag{15}$$

```
> evalf(newton(f,a,n)-sqrt(2));
```

$$0. \tag{16}$$

evalf utilisé sans option donne une approximation avec 10 chiffres signicatifs, ici ce n'est pas suffisant. On tatonne un peu et on lui demande suffisamment de chiffres après la virgule:

```
> evalf(newton(f,a,n)-sqrt(2),802750);
```

$$4.5445127192\,10^{-802739} \tag{17}$$

On a une précision remarquable, en seulement 20 itérations et bien meilleure qu'avec les deux autres méthodes.
Remarque: on est dans un cas favorable ici et dans le cas général cette méthode demande des hypothèses plus fortes sur f (dérivabilité).