TABLEAUX DE SIGNES
Cas du produit de deux facteurs affines

Compétences mathématiques :

- Signe de $ax + b$. Tableau de signe.

Compétences informatiques :

- Tableau (ou matrice).

Prérequis :

- Écriture d'une procédure ; Chaînes de caractères ; Test « si alors sinon ».

Le problème

On veut étudier le signe de $f(x) \times g(x)$ avec f et g deux fonctions affines et en rentrant uniquement comme arguments f et g pour obtenir le tableau de signe correspondant.

Partie 1: Étude avec « papier-crayon »

1. Notre but est d'obtenir un tableau de ce style :

Valeurs de x		1		2	
Signe de $ax + b$	3	4	5	6	7
Signe de $cx + d$	8	9	10	11	12
Signe du produit	13	14	15	16	17

Reste à savoir comment remplir les 17 cases numérotées.

2. Reproduisez ce tableau au brouillon (sans les numéros !) et remplissez-le dans le cas particulier : $a = -2$, $b = -3$, $c = 4$ et $d = 5$.

3. Faites de même avec $a = -2$, $b = -3$, $c = -4$ et $d = 5$. Il reste maintenant à expliquer tout ça à notre ordinateur...

Partie 2: Étude avec XCAS

Le but du jeu

En tapant `signeproduit(x->-2*x+3,x->4*x+5)` on veut obtenir :

$$\begin{bmatrix} \text{valeurs de x} & & \frac{-5}{4} & & \frac{3}{2} & \\ \text{signe de -2x+3} & + & & + & 0 & - \\ \text{signe de 4x+5} & - & 0 & + & & + \\ \text{signe du produit} & - & 0 & + & 0 & - \end{bmatrix}$$

1. **Cases 1 et 2** : sachant que **XCAS** sait résoudre les équations avec `resoudre(équation,variable)`, sait déterminer le minimum d'un ensemble de nombres avec `min(x1,x2)` et le maximum avec `max(x1,x2)`, proposez une méthode pour calculer `xmin` que nous mettrons dans la case 1 et `xmax` que nous mettrons dans la case 2.

2. **Les tableaux** : pour créer un tableau de, par exemple, 2 lignes et 3 colonnes, on utilise des « liste de listes » : une ligne sera représentée par 3 nombres (comme 3 colonnes) séparés par des virgules entre crochets, la liste de ces lignes étant elle-même mise entre crochets, if you see what I mean...

Par exemple, pour créer :

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

on rentre :

```
[ [1,2,3],[4,5,6],[7,8,9] ]
```

Ainsi, notre procédure commencera par :

```
signeproduit2(f,g):={ // nom de la procédure et des paramètres
z1:=resoudre(f(x)=0)[0]; // le [0] indique qu'on prend le 1er élément de la liste des
    solutions.
z2:=resoudre(g(x)=0)[0]; // z comme... zéro
zmin:=min(z1,z2);
zmax:=max(z1,z2);
   [ ["valeurs de x","  ",zmin,"  ",zmax,"  "],

     ["signe de "+f(x),... // à vous de compléter le reste !
```

Vous vous rappelez qu'on met du « texte » (on parle plutôt de *chaîne de caractères*) entre guillemets et qu'on colle (on dit *concatène*) des caractères avec des `+`.

Ainsi en rentrant `"signe de "+f(x)`, on fera comprendre à **XCAS** qu'il doit afficher `signe de -2x+3` si f est $x \mapsto -2x+3$.

3. **Case 3** : on met un - ou un + au hasard avec une chance sur deux de se tromper ou bien on réfléchit...Comme c'est le début, je vous rappelle votre cours : que se passe-t-il avant la valeur d'annulation de f ? Vous vous souvenez du test « si...alors...sinon... » ? Ici, cela donne :

```
["signe de "+f(x),   si f(zmin-1)>0 alors " + "; sinon " - "; fsi, // le reste à compléter.
```

Justifiez le test choisi.

4. **Case 4** : ici, on met un zéro ou rien du tout. Comment le savoir ? N'oubliez pas que cette ligne concerne le facteur $f(x)$ et qu'on teste une égalité avec `==`. Comme Noël est encore proche, voici un nouveau cadeau :

```
si f(zmin)==0 alors 0; sinon " ";fsi,
```

5. **Case 5** : ici, on a besoin de tester sur un nombre pris entre zmin et zmax : prenons par exemple leur « milieu »...Dernier cadeau :

```
si f((zmin+zmax)*.5)>0 alors " + "; sinon " - "; fsi,
```

6. **Les cases de 6 à 12** : on revient à l'un des trois cas précédents en adaptant un peu.

7. **Case 13** : que pensez-vous de ce code :

```
si g(zmin-1)*f(zmin-1)>0 alors " + "; sinon " - "; fsi,
```

8. **Les cases de 14 à 17** : les cases 14 et 16 ne sont pas trop compliquées à remplir. Pour les cases 15 et 17, inspirez-vous de la case 13.

9. **Est-ce que ça marche ?** Il ne reste plus qu'à reprendre notre exemple :

```
signeproduit2(x->-2*x+3,x->(4*x+5))
```

Partie 3: Généralisation

On a un logiciel qui calcule comme un dieu et on se contente de lui demander le signe du produit de deux facteurs affines : ayons plus d'ambition !

Malheureusement, cela demande un peu de savoir faire informatique. Nous pouvons malgré tout relever quelques difficultés à défaut de vouloir les résoudre. Qu'est-ce qui cloche dans la procédure précédente si $f(x) = x^2 + 1$? Si $f(x) = x^2 - 1$? Si on a plus de deux facteurs ? Si deux facteurs ont un zéro en commun ?

Voici une procédure qui règle ces problèmes...mais qui est un peu plus compliquée que la précédente :

```
signeproduit(L):={
L:=apply(f->unapply(f,x),[L])// on transforme les expressions en fonctions
n:=size(L);
Z:=NULL;
pour k de 0 jusque n-1 faire // on crée la suite des zéros
```

```
  si size(resoudre(L[k](x)=0)>0) alors // Pour les expressions qui n'ont pas de zéros
                    pour j de 0 jusque size(resoudre(L[k](x)=0))-1 faire
                        Z:=Z,simplifier(resoudre(L[k](x)=0)[j]);
                    fpour;
  fsi;
fpour;
Z:=sort(Z); // on classe les zéros dans l'ordre croissant
nz:=size(Z);
pour u de 1 jusque nz-2 faire
     si Z[u]==Z[u+1] alors Z:=Z[0..u-1],Z[u+1..nz-1];nz:=nz-1; // pour les zéros en double
     fsi;
fpour;
nz:=size(Z);
l0:=NULL;li:=NULL;lr:=NULL; // on initialise nos suites
pour m de 0 jusque nz-1 faire l0:=l0,Z[m]," ";fpour; // on crée la suite des zéros (car c'était
     une liste)
 [ ["valeurs de x"," ",l0 ],
   pour p de 0 jusque n-1 faire lp:=NULL; // il y aura n lignes correspondant aux n expressions
       .
                                          // à chaque tour de boucle, on rajoute une ligne
     li:=li,["signe de "+L[p](x),   si L[p](Z[0]-1.0)>0 alors " + "; sinon " - "; fsi, // test
         pour le signe àgauche du premier zéro
                          pour r de 0 jusque nz-2 faire  // boucle pour le reste des colonnes
                            lp:=lp,si simplifier(L[p](Z[r]))==0 alors 0; sinon " ";fsi, //
                                test pour le zéro
                            si L[p]((Z[r]+Z[r+1])*.5)>0 alors " + "; sinon " - "; fsi // test
                                pour le signe entre deux zéros
                          fpour,
                          si simplifier(L[p](Z[nz-1]))==0 alors 0; sinon " ";fsi, // test du
                              dernier zéro
                          si L[p](Z[nz-1]+1.0)>0 alors " + "; sinon " - "; fsi // test du
                              dernier signe à droite du dernier zéro
     ];
    fpour,  // on passe à la dernière ligne avec le même type de tests
    ["signe du produit", si product(L[s](Z[0]-1.0),s,0,n-1)>0 alors " + "; sinon " - "; fsi,0,
         // on test les signes des produits cette fois
                        pour t de 0 jusque nz-2 faire
                          lr:=lr,si product(L[s]((Z[t]+Z[t+1])*.5),s,0,n-1)>0 alors " + ";
                             sinon " - "; fsi,0;
                        fpour,
                        si product(L[s](Z[nz-1]+1.0),s,0,n-1)>0 alors " + "; sinon " - "; fsi
     ]]}:;
```

Ce qui donne pour $(-2x+3)(-4x+5)(x^2-3)(x+1)(x^2-1)(\sqrt{x^2+1})(x(\cos(x)+2)$:

```
signeproduit(-2*x+3,-4*x+5,x^2-3,x+1,x^2-1,sqrt(x^2+1),x*(cos(x)+2)
```

valeurs de x		$-(\sqrt{3})$		-1		0		1		$\frac{5}{4}$		$\frac{3}{2}$		$\sqrt{3}$	
signe de -2*x+3	+		+		+		+		+		+	0	-		-
signe de -4*x+5	+		+		+		+		+	0	-		-		-
signe de x^2-3	+	0	-		-		-		-		-		-	0	+
signe de x+1	-		-	0	+		+		+		+		+		+
signe de x^2-1	+		+	0	-		-	0	+		+		+		+
signe de sqrt(x^2+1)	+		+		+		+		+		+		+		+
signe de x*(cos(x)+2)	-		-		-	0	+		+		+		+		+
signe du produit	+	0	-	0	-	0	+	0	-	0	+	0	-	0	+