

Un noyau de tableur

On veut écrire un tableur genre *excel* ou *oocalc*. On ne s'intéressera pas aux problèmes graphiques dans ce projet, mais le noyau doit être prévu pour être utilisé sans modification avec une interface quelconque (graphique ou non).

Une cellule (identifiée par un n° de ligne et un n° de colonne, par exemple *L4C5*) peut être soit vide, soit contenir une valeur numérique (entière), un texte, une *formule* ou une valeur d'erreur.

Pour simplifier, les formules seront simplement des sommes de valeurs de cellules ou de constantes, par exemple $L4C5 = L3C5 + L2C4 + 1$.

Dans les calculs des formules, les valeurs des cellules vides seront 0, et les cellules contenant des textes devront donner une valeur d'erreur.

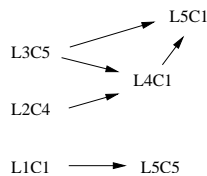
Le noyau du tableur devra bien évidemment, calculer les valeurs des cellules et afficher la feuille de calcul (sur demande). Par exemple pour un tableur 5×5 (? est affiché par le système pour demander une nouvelle commande):

```
? L1C1="XXX"
? L3C5=2
? L2C4=3
? L4C1=L3C5+L2C4+2
? L5C1=L3C5+L4C1
? L5C5=L1C1+1
? V (pour voir)
```

	1	2	3	4	5
1	XXX				
2				3	
3					2
4	7				
5	9				ERR

Le noyau devra être indépendant de la manière de saisir les formules et d'afficher la feuille de calcul.

Le noyau devra propager le changement de la valeur d'une cellule à toutes celles qui en dépendent (directement ou indirectement), et détecter les circularités dans les formules. Il est donc nécessaire que chaque cellule connaisse celles qui en dépendent : on a un graphe, qui sur l'exemple est :



Il est clair que, si la valeur de *L3C5* est modifiée, il faut recalculer *L4C1* avant *L5C1*. On fera donc un *tri topologique* pour déterminer l'ordre dans lequel

devront être recalculées les cellules, et on attachera une liste ordonnée à chaque cellule, par exemple pour $L3C5$, $\langle L4C1, L5C1 \rangle$. Ce tri permet aussi de détecter les circularités.

Si une formule change, le graphe doit bien sûr être recalculé *localement*, et les listes triées *concernées* doivent être reconstruites.

Enfin, ce noyau devra adapter les formules aux «copier-coller» : par exemple copier $L5C1$ en $L15C11$ devra donner à $L15C11$ la formule $L13C15 + L14C11$ et entraîner le calcul de sa valeur (ainsi que la modification *locale* du graphe et des listes triées *concernées*). Noter que, si de manière interne, les formules codent les cellules en coordonnées relatives à la cellule destination (par exemple $L5C1 = (-2, +4) + (-1, 0)$), les formules sont identiques.