Explications du programme

1 : On part du calcul de PGCD(*a*, *b*). Puisqu'on aura « plusieurs » *a* et *b*, mieux vaut tous les mettre dans des listes : les premiers *a* et *b* seront donc en fait *a*[1] et *b*[1]. Par exemple, *a*[3] sera en fait *q*[2] (le quotient de la précédente division euclidienne) et *b*[3] sera *r*[3]. Dès que ces nouvelles variables sont définies, on peut passer à la division euclidienne.

2 : On effectue la division euclidienne : mod(*a*, *b*) désigne la fonction qui retourne le reste dans la division euclidienne de *a* par *b*, et int(*a*/*b*) son quotient. Comme vu dans la leçon, $u_1 = 1$ et $v_1 = -q_1$ (en fait, c'est u_0 et v_0, mais la calculatrice ne tolère pas *u*[0] ou *v*[0], les « indices » doivent commencer à 1 !!). C'est donc ces variables que nous créons.

3 : Après une division euclidienne, on vérifie si le reste de la nouvelle division ne serait pas nul, auquel cas, on peut arrêter le programme, puisqu'on sait que le PGCD est le dernier reste **non nul** !! On passe alors directement à l'affichage de la réponse (via le `Goto`).

4 : Si ce fameux reste n'est pas nul, alors on fait une division euclidienne de plus. Comme le dit la leçon, on a alors $u_2 = -u_1q_2$ et $v_2 = 1 - v_1q_2$, éléments que nous allons définir ici.

5 : On fait alors une boucle `While` vérifiant en premier lieu si le prochain reste est nul ou non. Tant qu'il n'est pas nul, on fait des divisions euclidiennes, à l'issue desquels les variables u_i et v_i sont créées (pour le rang *i*), puis les variables a_{i+1} et b_{i+1} permettant de continuer les divisions euclidiennes.

6 : C'est uniquement l'affichage.

ATTENTION : **On va me dire, mais pourquoi n'a-t-on pas inséré l'étape 4 dans le For ??**
Tout simplement parce qu'à l'issue de la division euclidienne, on doit créer les variables u_2 et v_2. Mais ce n'est qu'à partir du rang 3 (d'où d'ailleurs le $3 \rightarrow i$ avant la boucle `While`) que les u_i et v_i suivent une relation de récurrence utilisable dans un `For`, pas avant !!