Un programme pour Bézout

1. La théorie.

a) L'algorithme d'Euclide.

On considère $a, b \in \mathbf{N}$ avec $b \neq 0$. On pose $a = r_0$, $b = r_1$. On effectue la division euclidienne de a par b : $a = bq+r$ avec $0 \leq r < b$. On pose $q = q_1$, $r = r_2$. On a donc $r_0 = r_1 q_1 + r_2$ avec $0 \leq r_2 < r_1$ et $d = pgcd\,(a, b) = pgcd\,(r_0, r_1) = pgcd\,(r_1, r_2)$.
On construit ainsi par récurrence des entiers $r_0, r_1, \cdots, r_k, r_{k+1}$ et $q_1, \cdots, q_k$ avec $r_{k-1} = q_k r_k + r_{k+1}$, $0 \leq r_{k+1} < r_k$ et $d = pgcd\,(r_k, r_{k+1})$. Si on a $r_{k+1} = 0$ on a $d = r_k$ et on s'arrête, sinon on continue l'algorithme en divisant r_k par r_{k+1}.
Comme on a $0 \leq r_{k+1} < r_k < \cdots < r_1$ on voit que l'on obtient nécessairement un reste nul au bout d'au plus r_1 opérations. Si on désigne par r_n le dernier reste non nul on a donc $r_{n-1} = q_n r_n$ d'où, $d = pgcd\,(r_{n-1}, r_n) = r_n$.

b) Le théorème de Bézout.

Pour montrer Bézout, on utilise l'algorithme d'Euclide. On va montrer, par récurrence sur k que, pour tout k avec $0 \leq k \leq n$, il existe des entiers $u_k, v_k \in \mathbf{Z}$ vérifiant $r_k = u_k a + v_k b$.
L'assertion est vraie pour $k = 0$ puisqu'on a $r_0 = a = 1 \times a + 0 \times b$ et pour $k = 1$ puisqu'on a $r_1 = b = 0 \times a + 1 \times b$. Supposons l'assertion prouvée pour tout entier $\leq k$, avec k fixé vérifiant $1 \leq k < n$, et montrons la pour $k + 1$. On a $r_{k+1} = r_{k-1} - q_k r_k = u_{k-1} a + v_{k-1} b - q_k (u_k a + v_k b)$ d'où la relation cherchée en posant $u_{k+1} = u_{k-1} - q_k u_k$ et $v_{k+1} = v_{k-1} - q_k v_k$.
Si on applique l'assertion au cas $k = n$, comme on a $r_n = d = pgcd\,(a, b)$, on obtient bien la relation de Bézout cherchée.
Pour écrire la récurrence sur les coefficients u_k, v_k le mieux est d'introduire la matrice m_k suivante : $m_k = \begin{pmatrix} u_{k-1} & v_{k-1} \\ u_k & v_k \end{pmatrix}$ car la relation de récurrence s'écrit alors $m_{k+1} = p_k m_k$ où p_k est la matrice $\begin{pmatrix} 0 & 1 \\ 1 & -q_k \end{pmatrix}$.

2. Le programme sur Voyage.

```
euclide(a,b)
Prgm
EffES
Local m,q,r
Identité(2) → m
While b > 0
mod(a,b) → r
(a-r)/b → q
b → a : r → b : [[0,1][1,-q]]*m → m
EndWhile
Disp "pgcd", a   Disp "u", m[1,1]   Disp "v", m[1,2]
EndPrgm
```