```
import numpy as np
from numpy.polynomial import polynomial
import math

def prime_sieve(N):
    is_prime = np.full(N, True)
    is_prime[:2] = False
    for p in range(2, math.isqrt(N) + 1):
        if is_prime[p]:
            is_prime[p*p::p] = False
    return np.nonzero(is_prime)[0]

n = 16
Premiers = prime_sieve(n - 1)[1:]
Complementaire = []
for k in Premiers:
    Complementaire.append(n-k)
print('ensemble des nombres premiers considere = ',Premiers)
p = polynomial.polyfromroots(Premiers)
q = polynomial.polyfromroots(Complementaire)
print(p)
print(q)
pp = list(reversed(p))
qq = list(reversed(q))
print(len(p))
sylvester = np.zeros((2*len(p)-2,2*len(p)-2),dtype='int')
for y in range(len(p)-1):
    for x in range(len(p)):
        sylvester[y,y+x] = pp[x]
        sylvester[y+len(p)-1,y+x] = qq[x]
print('sylvester = \n',sylvester)
print('la matrice est de determinant : ',np.linalg.det(sylvester))
```