

```

import sympy
from sympy import resultant, factor, symbols, expand
import numpy as np
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]

print('dans les resultats ci-apres, le nombre de decompositions de Goldbach
possible d un nombre pair,')
print('comme sommes faisant intervenir tous les nombres premiers jusqu a ce
nombre moins deux, est le degre du monome contenant le nombre pair en question
dans le developpement du polynome total.\n')
x, varn = symbols('x varn')
for n in range(6,102,2):
    print('\n',n)
    Premiers = prime_sieve(n - 1)[1:]
    polynome = 1
    lautrepolynome = 1
    for pk in Premiers:
        polynome = polynome * (x-pk)
        p = expand(polynome)
        lautrepolynome = lautrepolynome * ((varn-x)-pk)
        q = expand(lautrepolynome)
    print(factor(resultant(p,q)))

```