```
In[138]:= (** Author: Gilmar Rodriguez Pierluissi  **)

In[139]:= (** This Mathematica notebook was built with the purpose
           to clarifying the computations behind OEIS sequence A293858.  **)

In[140]:= (** EXAMPLE 1:  **)

In[141]:= n = 2707118;

In[142]:= m = n / 2

Out[142]= 1353559

In[143]:= PrimeQ[m]

Out[143]= False

In[144]:= goldbachpair[n_?EvenQ] /; n > 3 :=
           Block[{m = PrimePi[n / 2], p}, While[! PrimeQ[n - (p = Prime[m])], m--];
            {p, n - p}]

In[145]:= gpair = goldbachpair[n]

Out[145]= {1351837, 1355281}

In[146]:= p = gpair[[1]]

Out[146]= 1351837

In[147]:= (** When we know p = 1351837 beforehand then we can use p to calculate:  **)

In[148]:= L1 = PrimePi[m - 1] - PrimePi[p + 1]

Out[148]= 120

In[149]:= (** But, when you don't know p beforehand
           then you use the following Mathematica program:  **)

In[150]:= Clear[p]

In[151]:= PreviousPrime[n_] := NextPrime[n, -1]

In[152]:= L[n_?EvenQ] := Module[{m = n / 2}, If[PrimeQ[m], l = 0,
             l = Length[Drop[Most@NestWhileList[PreviousPrime, m, ! PrimeQ[n - #] &], 1]]];
            l]

In[153]:= L2 = L[n]

Out[153]= 120

In[154]:= (** These are the primes p that we are dealing with:  **)
```

In[155]:=
```
pvalues = Drop[NestList[PreviousPrime, m, L2], 1]
```

Out[155]=
```
{1 353 551, 1 353 487, 1 353 479, 1 353 463, 1 353 449, 1 353 433, 1 353 397, 1 353 383, 1 353 377,
 1 353 371, 1 353 329, 1 353 311, 1 353 301, 1 353 293, 1 353 281, 1 353 277, 1 353 269, 1 353 259,
 1 353 257, 1 353 241, 1 353 239, 1 353 223, 1 353 221, 1 353 197, 1 353 179, 1 353 173, 1 353 137,
 1 353 133, 1 353 101, 1 353 091, 1 353 089, 1 353 059, 1 353 043, 1 353 029, 1 353 019, 1 353 007,
 1 352 999, 1 352 993, 1 352 987, 1 352 977, 1 352 969, 1 352 963, 1 352 957, 1 352 921, 1 352 917,
 1 352 903, 1 352 893, 1 352 881, 1 352 873, 1 352 863, 1 352 861, 1 352 849, 1 352 839, 1 352 807,
 1 352 803, 1 352 783, 1 352 779, 1 352 777, 1 352 773, 1 352 761, 1 352 753, 1 352 749, 1 352 669,
 1 352 657, 1 352 641, 1 352 627, 1 352 599, 1 352 597, 1 352 557, 1 352 543, 1 352 521, 1 352 489,
 1 352 459, 1 352 447, 1 352 443, 1 352 441, 1 352 419, 1 352 389, 1 352 383, 1 352 371,
 1 352 369, 1 352 359, 1 352 347, 1 352 317, 1 352 311, 1 352 293, 1 352 291, 1 352 279,
 1 352 269, 1 352 257, 1 352 227, 1 352 209, 1 352 207, 1 352 203, 1 352 201, 1 352 191,
 1 352 171, 1 352 167, 1 352 149, 1 352 123, 1 352 119, 1 352 111, 1 352 107, 1 352 093,
 1 352 069, 1 351 997, 1 351 991, 1 351 981, 1 351 979, 1 351 967, 1 351 957, 1 351 949,
 1 351 921, 1 351 919, 1 351 913, 1 351 901, 1 351 897, 1 351 853, 1 351 843, 1 351 841}
```

In[156]:=
```
(** These are the n-p values that we are dealing with:  **)
```

In[157]:=
```
nminusp = n - pvalues
```

Out[157]=
```
{1 353 567, 1 353 631, 1 353 639, 1 353 655, 1 353 669, 1 353 685, 1 353 721, 1 353 735, 1 353 741,
 1 353 747, 1 353 789, 1 353 807, 1 353 817, 1 353 825, 1 353 837, 1 353 841, 1 353 849, 1 353 859,
 1 353 861, 1 353 877, 1 353 879, 1 353 895, 1 353 897, 1 353 921, 1 353 939, 1 353 945, 1 353 981,
 1 353 985, 1 354 017, 1 354 027, 1 354 029, 1 354 059, 1 354 075, 1 354 089, 1 354 099, 1 354 111,
 1 354 119, 1 354 125, 1 354 131, 1 354 141, 1 354 149, 1 354 155, 1 354 161, 1 354 197, 1 354 201,
 1 354 215, 1 354 225, 1 354 237, 1 354 245, 1 354 255, 1 354 257, 1 354 269, 1 354 279, 1 354 311,
 1 354 315, 1 354 335, 1 354 339, 1 354 341, 1 354 345, 1 354 357, 1 354 365, 1 354 369, 1 354 449,
 1 354 461, 1 354 477, 1 354 491, 1 354 519, 1 354 521, 1 354 561, 1 354 575, 1 354 597, 1 354 629,
 1 354 659, 1 354 671, 1 354 675, 1 354 677, 1 354 699, 1 354 729, 1 354 735, 1 354 747,
 1 354 749, 1 354 759, 1 354 771, 1 354 801, 1 354 807, 1 354 825, 1 354 827, 1 354 839,
 1 354 849, 1 354 861, 1 354 891, 1 354 909, 1 354 911, 1 354 915, 1 354 917, 1 354 927,
 1 354 947, 1 354 951, 1 354 969, 1 354 995, 1 354 999, 1 355 007, 1 355 011, 1 355 025,
 1 355 049, 1 355 121, 1 355 127, 1 355 137, 1 355 139, 1 355 151, 1 355 161, 1 355 169,
 1 355 197, 1 355 199, 1 355 205, 1 355 217, 1 355 221, 1 355 265, 1 355 275, 1 355 277}
```

In[158]:=
```
(** Verify that the n-p values are not prime:  **)
```

In[159]:=
```
PrimeQ[nminusp]
```

Out[159]=
```
{False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False, False, False, False}
```

In[160]:=
```
(** Now look at the following value:  **)
```

In[161]:=
```
p = PreviousPrime[Last[pvalues]]
```
Out[161]= 1 351 837

In[162]:=
```
PrimeQ[p]
```
Out[162]= True

In[163]:=
```
q = (n - p)
```
Out[163]= 1 355 281

In[164]:=
```
PrimeQ[q]
```
Out[164]= True

In[165]:=
```
p + q == n
```
Out[165]= True

In[166]:=
```
{p, q} == gpair
```
Out[166]= True

In[167]:=
```
(*****************************************************************************
 ******************************************************************************
 *****)
(** EXAMPLE 2:  **)
(** Here we use the above L[n_?EvenQ]
   function to build the values {n,L} for 4 ≤ n ≤ 148 :  **)
```

In[168]:=
```
nandLvalues = Table[{n, L[n]}, {n, 16, 148, 2}]
```
Out[168]=
```
{{16, 1}, {18, 0}, {20, 0}, {22, 0}, {24, 0}, {26, 0}, {28, 1}, {30, 0}, {32, 0}, {34, 0},
 {36, 0}, {38, 0}, {40, 1}, {42, 0}, {44, 2}, {46, 0}, {48, 1}, {50, 1}, {52, 0}, {54, 0},
 {56, 1}, {58, 0}, {60, 0}, {62, 0}, {64, 2}, {66, 1}, {68, 0}, {70, 1}, {72, 0}, {74, 0},
 {76, 2}, {78, 0}, {80, 0}, {82, 0}, {84, 0}, {86, 0}, {88, 1}, {90, 0}, {92, 3}, {94, 0},
 {96, 1}, {98, 3}, {100, 0}, {102, 1}, {104, 1}, {106, 0}, {108, 1}, {110, 2}, {112, 0},
 {114, 0}, {116, 2}, {118, 0}, {120, 0}, {122, 0}, {124, 2}, {126, 1}, {128, 0}, {130, 1},
 {132, 0}, {134, 0}, {136, 3}, {138, 0}, {140, 0}, {142, 0}, {144, 0}, {146, 0}, {148, 4}}
```

In[169]:=
```
(** The set of successive maxima values of the above
 set nandLvalues can be obtained using the following rule:  **)
```

In[170]:=
```
rule = {x___, {a_, b_}, {c_, d_}, {e_, f_}, y___} /;
    (d - b) / (c - a) < (f - b) / (e - a) :> {x, {a, b}, {e, f}, y};
```

In[171]:=
```
nandLvalues2 = nandLvalues //. rule
```
Out[171]= {{16, 1}, {44, 2}, {92, 3}, {148, 4}}

In[172]:=
```
(** The set of successive maxima values is abbreviated as follows:  **)
```

In[173]:=
```
Table[nandLvalues2[[i]][[1]], {i, 1, Length[nandLvalues2]}]
```
Out[173]= {16, 44, 92, 148}

In[174]:=
```
(*****************************************************************************
 ****************************************************************************
 *****)
```

In[175]:=
```
(** The complete Mathematica code to generate
 the successive maxima for sequence A293858 is:  **)
```

In[176]:=
```
PreviousPrime[n_] := NextPrime[n, -1]
```

In[177]:=
```
L[n_?EvenQ] := Module[{m = n / 2}, If[PrimeQ[m], l = 0,
   l = Length[Drop[Most@NestWhileList[PreviousPrime, m, ! PrimeQ[n - #] &], 1]]];
  l]
```

In[178]:=
```
f[n_] := For[m = n / 2, True, m--, Return[L[n]]];
For[n = 16;
 max = -1, True, n += 2, If[f[n] > max, Print[n];
  max = f[n]]]
```

```
16
44
92
148
368
400
530
688
992
1052
2228
3562
4952
7102
10 262
20 684
37 052
52 394
61 456
62 828
80 144
224 648
236 476
251 806
360 524
362 534
742 856
1 655 152
1 872 236
2 108 282
2 319 728
2 707 118
```

Out[178]= $Aborted