All about k-tuples

Calculations and coding by Matt C Anderson

January 2022

Contact matthewcharlesanderson2@gmail.com

These are k-tuples. They involve sets of prime numbers without repition.
<joke> They are not coffee. (Kurig k_cups) </end joke>.

These k-tuples are similar to prime constellations

See https://mattanderson.fun/f/prime-constellations

This webpage is paid for by me, and rented from an internet service provider (ISP).

Also, thanks to Norman Luhn and his webpage http://www.pzktupel.de/ktuplets.htm

Here is a Maple procedure for pairs of prime numbers. 2-tuples, if you will agree with me, are awesome.

```
> #This is Maple code
> # by Matt  Anderson
> # have procecure to search for and find  pairs of prime numbers
>
> #input difference between the two prime numbers in variable diff1
      (meaning difference 1)
>
> # note the two prime numbers are not neccessarily consecutive
      primes.
>
> # you could use ithprime() to optimise this code, and make the
      calculations sub microsecond.
> # but it is pretty fast already.  It is nice clean simple code.
>
>
>
>
```

```
mattsPrimePairsProc := proc(diff1);
 local a, counter, searchstop;
counter := 1 :
searchstop := 1000;                                  # local parameter
for a from 3 to searchstop by 2 do
if isprime(a) and isprime(a + diff1) then print(counter,
      "hourah!! we found a pair", a, " ", a + diff1);
 counter := counter + 1;
 end if;
 end do;
 end proc;
```

```
mattsPrimePairsProc := proc(diff1)
    local a, counter, searchstop;
    counter := 1;
    searchstop := 1000;
    for a from 3 by 2 to searchstop do
        if isprime(a) and isprime(a + diff1) then
            print(counter, "hourah!! we found a pair", a, " ", a + diff1)
            counter := counter + 1
        end if
    end do
end proc
```

```
> # now try the procedure - like a test drive for computer code.
```

```
> # note OEIS.org only records prime pairs for even numbers 2
      through 44.
> # we want original calculations, not in a public database.
> # calculate pairs for 46,48,50,52.54.100, and 1000.
> mattsPrimePairsProc(46)
                    1, "hourah!! we found a pair", 7, " ", 53
                    2, "hourah!! we found a pair", 13, " ", 59
                    3, "hourah!! we found a pair", 37, " ", 83
                    4, "hourah!! we found a pair", 43, " ", 89
                    5, "hourah!! we found a pair", 61, " ", 107
                    6, "hourah!! we found a pair", 67, " ", 113
                    7, "hourah!! we found a pair", 103, " ", 149
                   39, "hourah!! we found a pair", 967, " ", 1013
> #yeaaaaaah   success
> # and that,  gentlemen, is how you do that
> # quote from the awesome  movie  "Apollo 13", based on a real life
      near failure of a spacecraft.
> # Matt
> #share
>
> #now that we know that it works, here is another data set
> mattsPrimePairsProc(48)
                    1, "hourah!! we found a pair", 5, " ", 53
                    2, "hourah!! we found a pair", 11, " ", 59
                    3, "hourah!! we found a pair", 13, " ", 61
                    4, "hourah!! we found a pair", 19, " ", 67
                    5, "hourah!! we found a pair", 23, " ", 71
                    6, "hourah!! we found a pair", 31, " ", 79
                    7, "hourah!! we found a pair", 41, " ", 89
                    8, "hourah!! we found a pair", 53, " ", 101
                    9, "hourah!! we found a pair", 59, " ", 107
                   10, "hourah!! we found a pair", 61, " ", 109
                   11, "hourah!! we found a pair", 79, " ", 127
                   12, "hourah!! we found a pair", 83, " ", 131
                   13, "hourah!! we found a pair", 89, " ", 137
                   14, "hourah!! we found a pair", 101, " ", 149
```

Enjoy.

Have a nice day.

k-tuples for k in the set {2,3,4,5,6,7,8,9,10,11}.

2-tuples (pairs of prime numbers) through 11-tuples are possible with the Maple code below. See code block

```
> # input 0 for less than a 10 tuple.  Just give pattern.
>
>
tuple10 := proc(searchstop, diff1, diff2, diff3, diff4, diff5, diff6,
    diff7, diff8, diff9, diff10);
 local a, counter;
counter := 1;

    # passed parameter for list lengthsis searchstop
for a from 3 to searchstop by 2 do
if isprime(a) and isprime(a + diff1) and isprime(a + diff2)
    and isprime(a + diff3) and isprime(a + diff4) and isprime(a
    + diff5) and isprime(a + diff6) and isprime(a + diff7)
    and isprime(a + diff8) and isprime(a + diff9) and isprime(a
    + diff10) then
print(counter, "hourah!! we found a set", a, " ", a + diff1, " ", a + diff2,
    a + diff3, a + diff4, a + diff5, a + diff6, a + diff7, a + diff8, a
    + diff9, a + diff10);
 counter := counter + 1;
 end if;
 end do;
 end proc;
```

```
tuple10 := proc(searchstop, diff1, diff2, diff3, diff4, diff5, diff6,
    diff7, diff8, diff9, diff10)
    local a, counter;
    counter := 1;
    for a from 3 by 2 to searchstop do
        if isprime(a) and isprime(a + diff1) and isprime(a + diff2)
        and isprime(a + diff3) and isprime(a + diff4) and
        isprime(a + diff5) and isprime(a + diff6) and isprime(a
        + diff7) and isprime(a + diff8) and isprime(a + diff9) and
        isprime(a + diff10) then
            print(counter, "hourah!! we found a set", a, " ", a + diff1,
            " ", a + diff2, a + diff3, a + diff4, a + diff5, a + diff6, a
            + diff7, a + diff8, a + diff9, a + diff10);
            counter := counter + 1
        end if
    end do
end proc
```

```
> # now try the procedure - like a test drive for computer code.  Let's
      see if it works.
>
> # these have set repitition.  Pretty cool.  Good fun.
> tuple10(200, 12, 14, 0, 0, 0, 0, 0, 0, 0, 2);
                              1, "hourah!! we found a set", 5, " ", 17, " ", 19, 5, 5, 5, 5, 5, 5, 5, 7

                            2, "hourah!! we found a set", 17, " ", 29, " ", 31, 17, 17, 17, 17, 17, 17, 17,
                              19

                            3, "hourah!! we found a set", 29, " ", 41, " ", 43, 29, 29, 29, 29, 29, 29, 29,
                              31

                            4, "hourah!! we found a set", 59, " ", 71, " ", 73, 59, 59, 59, 59, 59, 59, 59,
                              61

                            5, "hourah!! we found a set", 137, " ", 149, " ", 151, 137, 137, 137, 137,
                              137, 137, 137, 139

                            6, "hourah!! we found a set", 179, " ", 191, " ", 193, 179, 179, 179, 179,
                              179, 179, 179, 181

> # a shorter list because I required a twin pair with pattern (0,2,12,
      14).  A 4-tuple, if you will.
>
>
> #Please spread this or similar code around.   Let me know, if you do.
> #email matthewcharlesanderson2@gmail.com
>
> # note OEIS.org only records prime pairs for even numbers 2
      through 44.

> # Also, OEIS has limited 3-tuples in the database.  They are deemed
      'not of general intrest'.

> tuple10(900, 50, 2, 0, 0, 0, 0, 0, 0, 0, 0)
                               1, "hourah!! we found a set", 3, " ", 53, " ", 5, 3, 3, 3, 3, 3, 3, 3, 3

                            2, "hourah!! we found a set", 11, " ", 61, " ", 13, 11, 11, 11, 11, 11, 11, 11,
                              11

                            3, "hourah!! we found a set", 17, " ", 67, " ", 19, 17, 17, 17, 17, 17, 17, 17,
                              17

                            4, "hourah!! we found a set", 29, " ", 79, " ", 31, 29, 29, 29, 29, 29, 29, 29,
                              29
```

5, "hourah!! we found a set", 59, " ", 109, " ", 61, 59, 59, 59, 59, 59, 59, 59

6, "hourah!! we found a set", 101, " ", 151, " ", 103, 101, 101, 101, 101, 101, 101, 101, 101

7, "hourah!! we found a set", 107, " ", 157, " ", 109, 107, 107, 107, 107, 107, 107, 107

8, "hourah!! we found a set", 149, " ", 199, " ", 151, 149, 149, 149, 149, 149, 149, 149

```
> # all prime numbers
> # We want original calculations, that are not already  in a public
      database.
>
> # now share on web.  (wheather it wants it or not :-)
> # have a nice day
> # Matthew
>
> # This 3-tuple has pattern (0,12,14).  It is a shorter list.  All the
      primes are  < 2,000.
> tuple10(500, 44, 0, 0, 0, 0, 0, 0, 0, 0);
```

1, "hourah!! we found a set", 3, " ", 47, " ", 3, 3, 3, 3, 3, 3, 3, 3, 3

2, "hourah!! we found a set", 17, " ", 61, " ", 17, 17, 17, 17, 17, 17, 17, 17, 17

3, "hourah!! we found a set", 23, " ", 67, " ", 23, 23, 23, 23, 23, 23, 23, 23, 23

4, "hourah!! we found a set", 29, " ", 73, " ", 29, 29, 29, 29, 29, 29, 29, 29, 29

5, "hourah!! we found a set", 53, " ", 97, " ", 53, 53, 53, 53, 53, 53, 53, 53, 53

6, "hourah!! we found a set", 59, " ", 103, " ", 59, 59, 59, 59, 59, 59, 59, 59, 59

7, "hourah!! we found a set", 83, " ", 127, " ", 83, 83, 83, 83, 83, 83, 83, 83, 83

8, "hourah!! we found a set", 107, " ", 151, " ", 107, 107, 107, 107, 107, 107, 107, 107, 107

This code needs work. Matt