```
> primorial := proc(n :: integer) :: integer;
    description "Return the product of the first n primes.  Find n#"
    local a, b;
    b := 1;
    for a from 1 to n do
    b := b·ithprime(a);
    end do;
    b;
    end proc:
> 
> c := primorial(4)
```

$$c := 210 \tag{1}$$

```
> nenp := proc(n :: integer) :: integer;
    description "Return the number of elements in the set of units mod n primorial"
    local a, b;
    b := 1;
    for a from 1 to n do
    b := b·(ithprime(a) − 1);
    end do;
    b;
    end proc:
> nenp(2)
```

$$2 \tag{2}$$

```
> nenp(3)
```

$$8 \tag{3}$$

```
> 
> 
> Unpv := proc(n :: integer) :: Vector;
    description "Return a Vector containing the units mod n primorial";
    local a, b, temp;
    temp := Vector[row](nenp(n));
    b := 1;
    for a from 1 to primorial(n) do
    if gcd(a, primorial(n)) = 1 then temp[b] := a; b := b + 1; end if;
    end do;
    temp;
    end proc:
> d := Unpv(3)
```

$$d := \begin{bmatrix} 1 & 7 & 11 & 13 & 17 & 19 & 23 & 29 \end{bmatrix} \tag{4}$$

```
> with(ArrayTools) :
> 
> patt := [0, 2, 6];
```

$$patt := [0, 2, 6] \tag{5}$$

```
> Size(patt)
```

$$\begin{bmatrix} 1 & 3 \end{bmatrix} \tag{6}$$

```
> offsets := proc(n :: integer, pat :: list) :: list;
```

```
        description "Return a set containing the offsets for a given pattern in the set of units mod n
        primorial";
     local a, b, counter, Unp, offs, sizepat;
     Unp := Unpv(n);
     offs := { };
    with(ArrayTools) :
     sizepat := Size(pat);
     for a from 1 to nenp(n) − sizepat[2] do
       b := 1;
       while b > 0 and b < sizepat[2] do
     if  Unp[a + b] − Unp[a] = pat[b + 1]  then
         b := b + 1;
       else b := 0; end if;
     end do;
       if b ≥ sizepat[2] − 1 then offs := offs union {Unp[a]}; end if;
       end do;
       [offs[ ]];
       end proc:
```

> `b := offsets(3, patt)`

$$b := [11, 17] \tag{7}$$

> `a`

$$a \tag{8}$$

> # so to find the 3 tuple [0,2,6] look at numbers 3#·n + {11,17}
> `patt := [0, 4, 6, 10, 12, 16];`

$$patt := [0, 4, 6, 10, 12, 16] \tag{9}$$

> `c := offsets(4, patt)`

$$c := [97] \tag{10}$$

> `c := offsets(5, patt)`

$$c := [97, 937, 1147, 1357, 2197] \tag{11}$$

> # so to find the 6 tuple [0,4,6,10,12,16], look at numbers 5#·n + {97, 937, 1147, 1357, 2197}
> `n := 3; pat := [0, 2, 6];`

$$n := 3$$
$$pat := [0, 2, 6] \tag{12}$$

> `d := [c[ ]]`

$$d := [97, 937, 1147, 1357, 2197] \tag{13}$$

>
```
refiner := proc(n :: integer, pat :: list, offs :: list) :: list;
    description "Given a pattern, offsets, and n; find the offsets for n+1";
    local a, b, nv, ns, sizeoffs, sizepat, sizenv;
    with(ArrayTools);
    sizeoffs := Size(offs);
    sizenv := ithprime(n + 1)·sizeoffs[2];
    nv := Vector[row](sizenv);
    for a from 1 to ithprime(n + 1) do
      for b from 1 to sizeoffs[2] do
      nv[b + sizeoffs[2]·(a − 1)] := offs[b] + primorial(n)·(a − 1);
    end do;
    end do;
    nv;
```

```
        sizepat := Size(pat);
        for a from 1 to sizenv do
         for b from 1 to sizepat[2] do
          if gcd(nv[a] + pat[b], ithprime(n + 1)) ≠ 1 then nv[a] := 0 end if;
        end do;
        end do;
        ns := convert(nv, set) minus {0};
        nv := [ns[ ]];
        nv;
        end proc:
```

> $d := refiner(4, patt, c)$;

$$d := [97, 937, 1147, 1357, 2197, 2407, 3247, 3457, 3667] \qquad \text{(14)}$$

> *# so the expression to use for the pattern [0,4,6,10,12,16] are #5·n + [97,937,1147,1357,2197]*

> $patt := [0, 6, 10, 12, 16, 22, 24, 30, 34, 36, 40, 42]$

$$patt := [0, 6, 10, 12, 16, 22, 24, 30, 34, 36, 40, 42] \qquad \text{(15)}$$

> $mult5pat12a := offsets(5, patt)$

$$mult5pat12a := [997] \qquad \text{(16)}$$

> $mult6pat12a := refiner(5, patt, mult5pat12a)$

$$mult6pat12a := [10237, 14857, 24097] \qquad \text{(17)}$$

> **save** $mult6pat12a$, "c:\Maplecode\mult6pat12a.m"

> $writedata($"c:\Maplecode\mult6pat12a.txt", $mult6pat12a)$;

> $with(ExcelTools)$:

> $Export(Vector(mult5pat12a),$ "c:/Maplecode/mult5pat12a.xls")

> $Export(Vector(mult6pat12a),$ "c:/Maplecode/mult6pat12a.xls")

> $mult7pas12a := refiner(6, patt, mult6pat12a)$

$$mult7pas12a := [10237, 14857, 24097, 54127, 70297, 104947, 134977, 144217, 174247, \qquad \text{(18)}$$
$$195037, 220447, 234307, 250477, 285127, 324397, 340567, 345187, 370597, 384457,$$
$$430657, 495337]$$

> $Export(Vector(mult7pas12a),$ "c:/Maplecode/mult7pas12a.xls")

> $mult8pat12a := refiner(7, patt, mult7pas12a)$

$$mult8pat12a := [14857, 24097, 104947, 134977, 234307, 324397, 340567, 384457, 654727, \qquad \text{(19)}$$
$$705547, 760987, 795637, 851077, 881107, 894967, 1005847, 1155997, 1165237, 1306147,$$
$$1366207, 1451677, 1516357, 1541767, 1585657, 1726567, 1751977, 1816657, 1876717,$$
$$1902127, 2052277, 2066137, 2096167, 2112337, 2146987, 2262487, 2366437, 2387227,$$
$$2472697, 2562787, 2606677, 2622847, 2657497, 2726797, 2876947, 2893117, 2937007,$$
$$3077917, 3087157, 3133357, 3207277, 3237307, 3297367, 3403627, 3447517, 3558397,$$
$$3588427, 3747817, 3807877, 3824047, 3858697, 4098937, 4219057, 4228297, 4304527,$$
$$4318387, 4334557, 4369207, 4429267, 4579417, 4604827, 4648717, 4699537, 4729567,$$
$$4789627, 4845067, 4918987, 4939777, 4965187, 5115337, 5159227, 5175397, 5240077,$$
$$5300137, 5325547, 5445667, 5475697, 5489557, 5535757, 5685907, 5720557, 5789857,$$
$$5810647, 5940007, 5986207, 6046267, 6140977, 6150217, 6300367, 6360427, 6411247,$$
$$6466687, 6510577, 6556777, 6651487, 6660727, 6870937, 6887107, 6981817, 7157377,$$
$$7171237, 7201267, 7282117, 7291357, 7397617, 7432267, 7642477, 7727947, 7792627,$$

7801867, 7852687, 8002837, 8028247, 8152987, 8178397, 8222287, 8312377, 8342407, 8363197, 8388607, 8538757, 8598817, 8663497, 8693527, 8748967, 8783617, 8899117, 8912977, 9003067, 9109327, 9213277, 9294127, 9363427, 9409627, 9439657, 9513577, 9529747, 9573637]

> $Export(Vector(mult8pat12a), \text{"c:/Maplecode/mult8pat12a.xls"})$

> $mult9pat12a := refiner(8, patt, mult8pat12a) :$

> $Export(Vector(mult9pat12a), \text{"c:/Maplecode/mult9pat12a.xls"})$

> $mult10pat12a := refiner(9, patt, mult9pat12a) :$

> $Export(Vector(mult10pat12a), \text{"c:/Maplecode/mult10pat12a.xls"})$

> $mult11pat12a := refiner(10, patt, mult10pat12a) :$

> $Export(Vector(mult11pat12a), \text{"c:/Maplecode/mult11pat12a.xls"})$

Error, (in ExcelTools:-Export) Export failed. Verify that the exported table does not contain more than 255 columns and 65535 rows.

> $with(ArrayTools) :$

> $Size(mult11pat12a)$

$$\begin{bmatrix} 1 & 522291 \end{bmatrix} \tag{20}$$

> $Export(Vector[row](mult11pat12a), \text{"c:/Maplecode/mult11pat12a.xls"})$

Error, (in ExcelTools:-Export) Export failed. Verify that the exported table does not contain more than 255 columns and 65535 rows.

> # Note the previous caluclation took 400 seconds

> $a := Vector(mult11pat12a) :$

> a

$$\begin{bmatrix} 1 .. 522291 \; Vector_{column} \\ Data\ Type:\ anything \\ Storage:\ rectangular \\ Order:\ Fortran\_order \end{bmatrix} \tag{21}$$

> $mult12pat12a := refiner(11, patt, mult11pat12a)$

$$[Length\ of\ output\ exceeds\ limit\ of\ 1000000] \tag{22}$$

>