Strings of Consecutive Happy Numbers
23 Feb 2008
1 Mar 2008
15 Mar 2008
21 Mar 2008
29 June 2009

The goal here is to show that the first string of nine consecutive happy numbers.

Note that Dr. Grundman uses S_2 but we will simply use S in the explanation, which in our program is the procedure 'onestep'.

Also note we will use a dot as the digit concatenation operator. Thus, 111112999994 could be written 11111.2.9999.94 if we wish.

First we define our procedures.

```
> restart;
> f := n → n^2;  #in case we ever want to investigate the cube of the digits, etc.
```

$$f := n \rightarrow n^2 \quad \textbf{(1)}$$

```
> bs := 10;
      #this is the base, in case we ever want to investigate binary or ternary or any other base.
```

$$bs := 10 \quad \textbf{(2)}$$

```
> onestep := proc(n1)
      #this is what Dr. Grundman calls S_2(n1) and what we will simply call S below.
  local ans, n, d;
  n := n1;
  ans := 0;
  while n > 0 do
  d := n mod bs;
  ans := ans + f(d);
  n := (n-d)/bs;
  end do;
  ans;
  end;
```

```
onestep := proc(n1)                                                    (3)
    local ans, n, d;
    n := n1;
    ans := 0;
    while 0 < n do d := mod(n, bs); ans := ans + f(d); n := (n - d)/bs end do;
    ans
end proc
```

```
> happy := proc(n)
      # returns -1 if not happy, and returns the number of steps to reach 1 if it is happy
  local m, j, height;
  m := n;
```

```
    height := -1;
    for j from 1 to 100 while (m > 1 and m ≠ 4) do
    m := onestep(m);
     end do;
    if m = 1 then height := j; end if;
    height;
    end;
happy := proc(n)                                                    (4)
    local m, j, height;
    m := n;
    height := -1;
    for j to 100 while 1 < m and m <> 4 do m := onestep(m) end do;
    if m = 1 then height := j end if;
    height
end proc
```

```
>
```

The next procedure is only needed when we want to find the smallest N with S(N) = n for a given n. A separate worksheet has the details on how this is constructed. The array contains the smallest N for 1 <= n <= 486 = 6*81.

```
> lowS := [1, 11, 111, 2, 12, 112, 1112, 22, 3, 13, 113, 222, 23, 123, 1123, 4, 14, 33, 133, 24,
      124, 233, 1233, 224, 5, 15, 115, 1115, 25, 125, 1125, 44, 144, 35, 135, 6, 16, 116, 1116,
      26, 45, 145, 335, 226, 36, 136, 1136, 444, 7, 17, 117, 46, 27, 127, 1127, 246, 227, 37, 137,
      1137, 56, 156, 1156, 8, 18, 118, 337, 28, 128, 356, 1356, 66, 38, 57, 157, 266, 238, 257,
      1257, 48, 9, 19, 119, 248, 29, 129, 1129, 466, 58, 39, 139, 1139, 258, 239, 1239, 448, 49,
      77, 177, 68, 168, 277, 1277, 268, 458, 59, 159, 666, 368, 259, 1259, 2666, 78, 178, 359,
      468, 69, 169, 1169, 2468, 269, 378, 577, 1577, 568, 369, 1369, 88, 188, 79, 179, 288, 469,
      279, 1279, 668, 388, 578, 379, 1379, 2388, 569, 1569, 488, 89, 189, 777, 1777, 289, 1289,
      2777, 4668, 588, 389, 579, 1579, 2588, 2389, 2579, 4488, 489, 99, 199, 688, 1688, 299,
      1299, 2688, 4588, 589, 399, 1399, 3688, 2589, 2399, 12399, 788, 499, 779, 1779, 689,
      1689, 2779, 12779, 2689, 3788, 599, 1599, 5688, 3689, 2599, 888, 1888, 789, 1789, 2888,
      4689, 699, 1699, 6688, 3888, 2699, 3789, 5779, 15779, 5689, 3699, 4888, 889, 1889, 799,
      1799, 2889, 4699, 2799, 12799, 5888, 3889, 5789, 3799, 13799, 23889, 5699, 15699,
      4889, 899, 1899, 6888, 16888, 2899, 12899, 26888, 45888, 5889, 3899, 5799, 15799,
      25889, 23899, 25799, 7888, 4899, 999, 1999, 6889, 16889, 2999, 12999, 26889, 37888,
      5899, 3999, 13999, 36889, 25899, 8888, 18888, 7889, 4999, 7799, 17799, 6899, 16899,
      27799, 38888, 26899, 37889, 5999, 15999, 56889, 36899, 25999, 8889, 18889, 7899,
      17899, 28889, 46899, 6999, 16999, 58888, 38889, 26999, 37899, 57799, 157799, 56899,
      36999, 48889, 8899, 18899, 7999, 17999, 28899, 46999, 27999, 127999, 58889, 38899,
      57899, 37999, 137999, 238899, 56999, 78888, 48899, 8999, 18999, 68889, 168889,
      28999, 128999, 268889, 378888, 58899, 38999, 57999, 157999, 258899, 88888, 188888,
      78889, 48999, 9999, 19999, 68899, 168899, 29999, 129999, 268899, 378889, 58999,
      39999, 139999, 368899, 258999, 88889, 188889, 78899, 49999, 77999, 177999, 68999,
      168999, 277999, 388889, 268999, 378899, 59999, 159999, 568899, 368999, 259999,
      88899, 188899, 78999, 178999, 288899, 468999, 69999, 169999, 588889, 388899,
      269999, 378999, 577999, 1577999, 568999, 369999, 488899, 88999, 188999, 79999,
      179999, 288999, 469999, 279999, 1279999, 588899, 388999, 578999, 379999, 1379999,
      888888, 569999, 788889, 488999, 89999, 189999, 688899, 1688899, 289999, 1289999,
      2688899, 3788889, 588999, 389999, 579999, 1579999, 2588999, 888889, 1888889,
      788899, 489999, 99999, 199999, 688999, 1688999, 299999, 1299999, 2688999, 3788899,
```

```
589999, 399999, 1399999, 3688999, 2589999, 888899, 1888899, 788999, 499999,
779999, 1779999, 689999, 1689999, 2779999, 3888899, 2689999, 3788999, 599999,
1599999, 5688999, 3689999, 2599999, 888999, 1888999, 789999, 1789999, 2888999,
4689999, 699999, 1699999, 5888899, 3888999, 2699999, 3789999, 5779999, 8888888,
5689999, 3699999, 4888999, 889999, 1889999, 799999, 1799999, 2889999, 4699999,
2799999, 12799999, 5888999, 3889999, 5789999, 3799999, 13799999, 8888889,
5699999, 7888899, 4889999, 899999, 1899999, 6888999, 16888999, 2899999, 12899999,
26888999, 37888899, 5889999, 3899999, 5799999, 15799999, 25889999, 8888899,
18888899, 7888999, 4899999, 999999 ] :
```

```
> minimalN := proc(n)
   local q, r, k, ans;
  global lowS;;
    if n < 487 then ans := lowS[n];
   else
  q := iquo(n, 81,'r');
   ans := lowS[n − (q − 5)·81]·10^(q − 5) + (10^(q − 5) − 1);
   end if;
   ans;
   end;
```

```
minimalN := proc(n)                                                    (5)
    local q, r, k, ans;
    global lowS;
    if n < 487 then
        ans := lowS[n]
    else
        q := iquo(n, 81, 'r'); ans := lowS[n − 81 * q + 405] * 10^(q − 5) + 10^(q − 5) − 1
    end if;
    ans
end proc
```

```
>
```

We claim the smallest sequence of 9 consecutive happy numbers begins with
N =
2699999999999999999999999999999999999999999999999999999999999999999999999999999999\
99999999999999999999999999999999999999999999999999999999799999999999999999999999\
999999999999999999999999999999999999999999999999999995
that is, 26 then 137 nines then the digit 7 then 74 nines then final digit 5.

We first show this does give nine in a row.

```
> N := 5 + 10·(10^74 − 1) + 7·10^75 + 10^76·(10^137 − 1) + 26·10^213
N :=                                                                   (6)
    26999999999999999999999999999999999999999999999999999999999999999999999999\
    99999999999999999999999999999999999999999999999999999999999999999799999999\
    9999999999999999999999999999999999999999999999999999999999999999995
```

```
> for j from -1 to 9 do print( j, happy(N + j) ); end do;
                                  -1, -1
                                   0, 6
                                   1, 7
```

$$2, 8$$
$$3, 5$$
$$4, 7$$
$$5, 8$$
$$6, 4$$
$$7, 5$$
$$8, 8$$
$$9, -1 \quad \textbf{(7)}$$

> $onestep(N); onestep(N+4);$

$$17261 \quad \textbf{(8)}$$

We first check for 9 in a row without a carry. Then N must have last digit 0 or 1.

```
> for a from 1 to 20000 do
  for d from 0 to 1 do
  c := 0;
  for j from 0 to 8 do
  if happy(a + onestep(d + j)) > 0 then c := c + 1; else c := 0; end if;
  if c > 5 then print(a, d, j, c) end if;
  end do; end do: end do:
```

$$3292, 0, 6, 6$$
$$3292, 1, 5, 6$$
$$11807, 0, 6, 6$$
$$11807, 1, 5, 6$$
$$12065, 0, 8, 6$$
$$12065, 1, 7, 6$$
$$14707, 0, 6, 6$$
$$14707, 1, 5, 6$$
$$17407, 0, 6, 6$$
$$17407, 1, 5, 6$$
$$18107, 0, 6, 6$$
$$18107, 1, 5, 6 \quad \textbf{(9)}$$

so we see there are not more than six in a row without a carry.

We look for 4 or 5 before the carry

```
> for a from 1 to 17500 do
   if happy(a + onestep(9)) > 0 and happy(a + onestep(8)) > 0 and happy(a + onestep(7))
      > 0 and happy(a + onestep(6)) > 0then print(a, happy(a + onestep(5)), happy(a
      + onestep(4))); end if;
  end do:
```

$$1839, -1, -1$$
$$3274, -1, 6$$
$$4806, -1, -1$$
$$8406, -1, -1$$
$$10839, -1, -1$$
$$11374, -1, -1$$
$$12680, 5, -1$$
$$14074, -1, -1$$

15038, -1, -1

17180, 5, -1 **(10)**

This also shows there are not six in a row before the carry.

Now for 4 or 5 after the carry; note the following shows there are not six in a row after the carry.

```
> for a from 1 to 17500 do
    if happy(a + onestep(0)) > 0 and happy(a + onestep(1)) > 0 and happy(a + onestep(2))
        > 0 and happy(a + onestep(3)) > 0 then print(a, happy(a + onestep(4)), happy(a
        + onestep(5))); end if;
  end do:
```

4554, -1, -1

5454, -1, -1

11201, -1, -1

11834, 6, -1

12101, -1, -1

12654, -1, -1

14561, -1, -1

14734, 6, -1

15461, -1, -1

16254, -1, -1

16966, -1, -1

17434, 6, -1 **(11)**

First we investigate the ones with five in a row before the carry to see if they extend to four after the carry.

```
> a := 12680;
  ktop := iquo(a, 81);
   for k from 0 to ktop do  # k is the number of nines at the end of N1.
  for d from 0 to 8 do  # d is the digit that precedes the k digits of nine at the end of N1.
  if happy(a + 2·d + 1 − k·81) > 0 and happy(a + 2·d + 1 − k·81 + onestep(1)) > 0
      and happy(a + 2·d + 1 − k·81 + onestep(2)) > 0  and happy(a + 2·d + 1 − k·81
       + onestep(3)) > 0 then print(a, k, d, happy(a + 2·d + 1 − k·81 + onestep(4)), a − d^2
       − k·81) end if;
  end do; end do:
```

$a := 12680$

$ktop := 156$ **(12)**

```
> a := 17180;
  ktop := iquo(a, 81);
   for k from 0 to ktop do  # k is the number of nines at the end of N1.
  for d from 0 to 8 do  # d is the digit that precedes the k digits of nine at the end of N1.
  if happy(a + 2·d + 1 − k·81) > 0 and happy(a + 2·d + 1 − k·81 + onestep(1)) > 0
      and happy(a + 2·d + 1 − k·81 + onestep(2)) > 0  and happy(a + 2·d + 1 − k·81
       + onestep(3)) > 0 then print(a, k, d, happy(a + 2·d + 1 − k·81 + onestep(4)), a − d^2
       − k·81) end if;
  end do; end do:
```

$a := 17180$

$ktop := 212$

17180, 74, 7, -1, 11137 **(13)**

Next we check the ones with five after the carry to see if they extend to four before the carry.

```
>  a ≔ 11834;
   ktop ≔ iquo(20000, 81);
    for k from 0 to ktop do  # k is the number of nines at the end of N1.
   for d from 0 to 8 do   # d is the digit that precedes the k digits of nine at the end of N1.
   q ≔ a − 2·d − 1 + k·81;
   if happy(q + onestep(9)) > 0 and happy(q + onestep(8)) > 0 and happy(q + onestep(7))
       > 0  and happy(q + onestep(6)) > 0 then print(a, k, d, happy(q + onestep(5)), q) end
      if;
   end do; end do:
```

```
                              a := 11834
                             ktop := 246                                   (14)
```

```
>  a ≔ 14734;
   ktop ≔ iquo(20000, 81);
    for k from 0 to ktop do  # k is the number of nines at the end of N1.
   for d from 0 to 8 do   # d is the digit that precedes the k digits of nine at the end of N1.
   q ≔ a − 2·d − 1 + k·81;
   if happy(q + onestep(9)) > 0 and happy(q + onestep(8)) > 0 and happy(q + onestep(7))
       > 0  and happy(q + onestep(6)) > 0 then print(a, k, d, happy(q + onestep(5)), q) end
      if;
   end do; end do:
```

```
                              a := 14734
                             ktop := 246                                   (15)
```

```
>  a ≔ 17434;
   ktop ≔ iquo(20000, 81);
    for k from 0 to ktop do  # k is the number of nines at the end of N1.
   for d from 0 to 8 do   # d is the digit that precedes the k digits of nine at the end of N1.
   q ≔ a − 2·d − 1 + k·81;
   if happy(q + onestep(9)) > 0 and happy(q + onestep(8)) > 0 and happy(q + onestep(7))
       > 0  and happy(q + onestep(6)) > 0 then print(a, k, d, happy(q + onestep(5)), q) end
      if;
   end do; end do:
```

```
                              a := 17434
                             ktop := 246                                   (16)
```

```
>  minimalN(11137); evalf(log10(%));
26999999999999999999999999999999999999999999999999999999999999999999999999999\
   99999999999999999999999999999999999999999999999999999999999999
                             138.4313638                                   (17)
```

None of these extend to four before the carry. Thus, the only candidate in this range is a= 17180, k=74, d=7. This gives
N =
26999999999999999999999999999999999999999999999999999999999999999999999999999999999\
   99999999999999999999999999999999999999999999999999999999.7.(74 *nines*).5

```
>  minimalN(17180 − 74·81 − 7^2)
26999999999999999999999999999999999999999999999999999999999999999999999999999\   (18)
   99999999999999999999999999999999999999999999999999999999999999
```

> $\frac{(\% + 1)}{10^{137}}$

$$27 \tag{19}$$

so there are 137 nines in the front part.