Strings of Consecutive Happy Numbers
23 Feb 2008
1 Mar 2008
15 Mar 2008
21 Mar 2008
1 Sept 2008
8 Sept 2008
27 Jun 2009
25 July 2009
28 July 2009

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

We claim the 603699 digit number $N = 288.(218491\ nines).3.(385203\ nines).3$ works.

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.
```

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

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

$$bs := 10 \quad (2)$$

```
> 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 603699 digit number $N = 288.(218491\ nines).3.(385203\ nines).3$ begins a string of 13 consecutive happy numbers.

We see this N has seven in a row before the carry, then six in a row after the carry.

```
> M1 := onestep(288) + 218491·9^2 + 3^2 + 385203·9^2;
   for j from 2 to 9 do print( j, happy(M1 + j^2) ); end do;
```

$M1 := 48899355$

2, -1

3, 7

4, 8

5, 8

6, 8

$$7, 6$$
$$8, 6$$
$$9, 7 \quad \textbf{(6)}$$

```
> M1 := onestep(288) + 218491·9^2 + (3 + 1)^2;
  for j from 0 to 6 do print( j, happy(M1 + j^2) ); end do;
```

$$M1 := 17697919$$
$$0, 8$$
$$1, 4$$
$$2, 4$$
$$3, 8$$
$$4, 7$$
$$5, 7$$
$$6, -1 \quad \textbf{(7)}$$

We first search for seven in a row before the carry, then seven after the carry. In our search for 12 in a row, we already saw that up to 13000000 there were not seven in a row.

```
> for a from 13000000 to 50000000 do
   if happy(a + 9^2) > 0 and happy(a + 8^2) > 0 and happy(a + 7^2) > 0 and happy(a + 6^2)
      > 0 and happy(a + 5^2) > 0 and happy(a + 4^2) > 0  and happy(a + 3^2) > 0
     then print(a, happy(a + 2^2), happy(a + 1^2) ); end if;
  end do:
```

$$48899355, -1, 7$$
$$48989355, -1, 7$$
$$48998355, -1, 7$$
$$49889355, -1, 7$$
$$49898355, -1, 7$$
$$49988355, -1, 7 \quad \textbf{(8)}$$

```
> for a from 13000000 to 50000000 do
   if happy(a) > 0 and happy(a + 1^2) > 0 and happy(a + 2^2) > 0 and happy(a + 3^2) > 0
     and happy(a + 4^2) > 0 and happy(a + 5^2) > 0 then print(a, happy(a + 6^2), happy(a
      + 7^2) ); end if;
  end do:
```

$$14459989, -1, -1$$
$$16779919, -1, 8$$
$$16797919, -1, 8$$
$$16799719, -1, 8$$
$$16977919, -1, 8$$
$$16979719, -1, 8$$
$$16997719, -1, 8$$
$$17299919, -1, 8$$
$$17679919, -1, 8$$
$$17697919, -1, 8$$
$$17699719, -1, 8$$
$$17769919, -1, 8$$

17796919, -1, 8
17799619, -1, 8
17929919, -1, 8
17967919, -1, 8
17969719, -1, 8
17976919, -1, 8
17979619, -1, 8
17992919, -1, 8
17996719, -1, 8
17997619, -1, 8
17999219, -1, 8
19279919, -1, 8
19297919, -1, 8
19299719, -1, 8
19677919, -1, 8
19679719, -1, 8
19697719, -1, 8
19729919, -1, 8
19767919, -1, 8
19769719, -1, 8
19776919, -1, 8
19779619, -1, 8
19792919, -1, 8
19796719, -1, 8
19797619, -1, 8
19799219, -1, 8
19927919, -1, 8
19929719, -1, 8
19967719, -1, 8
19972919, -1, 8
19976719, -1, 8
19977619, -1, 8
19979219, -1, 8
19992719, -1, 8
19997219, -1, 8
21799919, -1, 8
21979919, -1, 8
21997919, -1, 8
21999719, -1, 8
22559989, -1, -1
25259989, -1, -1
25599919, -1, 8

25959919, −1, 8
25995919, −1, 8
25999519, −1, 8
27199919, −1, 8
27919919, −1, 8
27991919, −1, 8
27999119, −1, 8
29179919, −1, 8
29197919, −1, 8
29199719, −1, 8
29559919, −1, 8
29595919, −1, 8
29599519, −1, 8
29719919, −1, 8
29791919, −1, 8
29799119, −1, 8
29917919, −1, 8
29919719, −1, 8
29955919, −1, 8
29959519, −1, 8
29971919, −1, 8
29979119, −1, 8
29991719, −1, 8
29995519, −1, 8
29997119, −1, 8
33699919, −1, 8
33969919, −1, 8
33996919, −1, 8
33999619, −1, 8
36399919, −1, 8
36939919, −1, 8
36993919, −1, 8
36999319, −1, 8
39369919, −1, 8
39396919, −1, 8
39399619, −1, 8
39639919, −1, 8
39693919, −1, 8
39699319, −1, 8
39936919, −1, 8
39939619, −1, 8
39963919, −1, 8

$$39969319, -1, 8$$
$$39993619, -1, 8$$
$$39996319, -1, 8$$
$$41459989, -1, -1$$
$$44159989, -1, -1$$
$$45888819, -1, 8$$
$$46688919, -1, 8$$
$$46689819, -1, 8$$
$$46698819, -1, 8$$
$$46868919, -1, 8$$
$$46869819, -1, 8$$
$$46886919, -1, 8$$
$$46889619, -1, 8$$
$$46896819, -1, 8$$
$$46898619, -1, 8$$
$$46968819, -1, 8$$
$$46986819, -1, 8$$
$$46988619, -1, 8$$
$$48588819, -1, 8$$
$$48668919, -1, 8$$
$$48669819, -1, 8$$
$$48686919, -1, 8$$
$$48689619, -1, 8$$
$$48696819, -1, 8$$
$$48698619, -1, 8$$
$$48858819, -1, 8$$
$$48866919, -1, 8$$
$$48869619, -1, 8$$
$$48885819, -1, 8$$
$$48888519, -1, 8$$
$$48896619, -1, 8$$
$$48966819, -1, 8$$
$$48968619, -1, 8$$
$$48986619, -1, 8$$
$$49668819, -1, 8$$
$$49686819, -1, 8$$
$$49688619, -1, 8$$
$$49866819, -1, 8$$
$$49868619, -1, 8$$
$$49886619, -1, 8 \quad \textbf{(9)}$$

> a;

$$13000001 \quad \textbf{(10)}$$

We see that there are not seven in a row after the carry, nor eight in a row before the carry, so the only way to get 13 in a row is to have a 7/6 split over the carry.

We test the smallest case which has seven before the carry to see if it can extend to six in a row after the carry.

First we see which digit will give us the smallest candidate. We see here our preference is d=8. The next bestg is d=3.

```
> iquo(48899355, 81)
```

$$603695 \tag{11}$$

```
> a := 48899355 :
  for d from 0 to 8 do
  m := a − 603690·9^2 − d^2;
  print(d, minimalN(m));
  end do:
```

0, 8888889
1, 13799999
2, 3889999
3, 2889999
4, 5689999
5, 4689999
6, 3788999
7, 3688999
8, 888889 **(12)**

We now see if we can get 6 in a row after the carry. We will just look when d=8 since this would be the optimal choice.

```
> a := 48899355;
  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 8 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·9^2;
  if happy(q + 0^2) > 0 and happy(q + 1^2) > 0 and happy(q + 2^2) > 0  and happy(q + 3^2)
      > 0 and happy(q + 4^2) > 0 and happy(q + 5^2) > 0 then print(a, k, d, happy(q
      + 6^2)) end if;
  end do; end do:
  print('fini');
```

a := 48899355
$ktop$:= 603695
$fini$ **(13)**

Since there are none for 8, let us try 3 next.

```
> a := 48899355;
  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 3 to 3 do  # d is the digit that precedes the k digits of nine at the end of N1.
  q := a + 2·d + 1 - k·9^2;
  if happy(q + 0^2) > 0 and happy(q + 1^2) > 0 and happy(q + 2^2) > 0  and happy(q + 3^2)
```

```
> 0 and happy(q + 4^2) > 0 and happy(q + 5^2) > 0 then print(a, k, d, happy(q
+ 6^2)) end if;
end do; end do:
print('fini');
```

$a := 48899355$

$ktop := 603695$

48899355, 283203, 3, −1

48899355, 359503, 3, −1

48899355, 381503, 3, −1

48899355, 385203, 3, −1

fini **(14)**

Bigger k is a smaller number, so the best one is k=385203. Thus, N must be 288.(218491 *nines*).3.(385203 *nines*).3 works.

> $48899355 - 603690 \cdot 81 - 3^2$; $minimalN(\%)$

456

2889999 **(15)**

> $603690 + 4 - 385203$

218491 **(16)**

> $3 + 218491 + 1 + 385203 + 1$

603699 **(17)**

>