Strings of Consecutive Happy Numbers
23 Feb 2008
1 Mar 2008
15 Mar 2008
21 Mar 2008
1 Sept 2008
6 Sept 2008
10 Sept 2008
12 Sept 2008
18 Sept 2008
19 Sept 2008
26 Sept 2008
9 Oct 2008

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

2222222222319612 digit number

15 in a row \( N = 77.(2222222222222220 \text{nines}) \cdot 3.(97388 \text{nines}) \cdot 3 \)

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.

```maple
restart;

f := n \rightarrow n^2;  # in case we ever want to investigate the cube of the digits, etc.

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

onestep := proc(n1)
    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;
end proc:
```

(1) 

(2) 

(3)
ans;
end;

onestep := proc(n1)
    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)
    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.

\begin{verbatim}
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) \cdot 81 \cdot 10^q - 5 + (10^{q - 5} - 1)];
  end if;
  ans;
end:

minimalbigN := 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 - 81 \cdot q + 405] \cdot 10^q (q - 5) + 10^{q - 5} (q - 5) - 1
  end if;
  ans;
end proc
\end{verbatim}

(6)
else
if $n < 1000$ then
    $q := \text{iquo}(n, 81, 'r');$
    $\text{ans} := \text{lowS}[n - (q - 5) \cdot 81] \cdot 10^q - 5 + (10^q - 5 - 1);$
else
    $q := \text{iquo}(n, 81, 'r');$
    $\text{ans} := [ \text{lowS}[n - (q - 5) \cdot 81], q - 5 ];$
end if; end if;
end;

minimalbigN := proc(n)
    local $q, r, k, \text{ans};$
    global \text{lowS};
    if $n < 487$ then
        $\text{ans} := \text{lowS}[n]$
    else
        if $n < 1000$ then
            $q := \text{iquo}(n, 81, 'r');$
            $\text{ans} := \text{lowS}[n - 81 \cdot q + 405] \cdot 10^{(q - 5)} + 10$
        else
            $q := \text{iquo}(n, 81, 'r');$
            $\text{ans} := [ \text{lowS}[n - 81 \cdot q + 405], q - 5 ]$
        end if
    end if
    $\text{ans}$
end proc

We first verify our candidate $N = 77.(2222222222222220 nines).3.(97388 nines).3$ works. First we check that the 7 before the carry work, then the 8 after the carry.

> for $j$ from 2 to 9 do
    print($j, \text{happy}(\text{onestep}(77) + 2222222222222220 \cdot 81 + 3^2 + 97388 \cdot 81 + j^2 ))$ end do;

2, 1
3, 7
4, 8
5, 8
6, 8
7, 6
8, 6
9, 7

> for $j$ from 0 to 8 do
    print($j, \text{happy}(\text{onestep}(77) + 2222222222222220 \cdot 81 + 4^2 + j^2 ))$ end do;

0, 5
1, 8
2, 6
3, 5
4, 5
We presume that the smallest example has a 7/8 or an 8/7 split over the carry.

We first try to find an 7/8 split over the carry.

The idea is that \( N = N_1d_0 \) where \( d_0 \) is the final digit. Let \( M_1 = S(N_1) \) and we divide \( M_1 \) into the last three digits and the preceding digits, call them \( M_2c \) where \( c \) is the final three digits. If \( c + d_0^2 < 1000 \) then \( S(S(N)) = S(M_2) + S(c + d_0^2) \) so we can merely check this to see if it is happy or not. Below we use \( b = S(M_2) \) and \( d_0 = i \) for the various final digits we are trying.

Recall that to guarantee that \( c + d_0^2 < 1000 \) we investigate \( c < 919 = 1000 - 9^2 \) for those "before the carry" and \( c < 1000 - 7^2 = 951 \) for those "after the carry."

\[
\begin{align*}
5, 5 \\
6, 6 \\
7, 6 \\
8, -1
\end{align*}
\]
Now we worry about possible carries in the M1 level when we calculated the 0,1,2,3,4,5,6,7 final digits.
(eight in a row after the carry.)
The problem is if \( c > 951 \) then the digit 6 could cause a carry into the M2 portion of M1.

```plaintext
> for b from 0 to 2000 do
  for c from 951 to 963 do  # so these have the last digit 7 carry but not any others.
    fini := false;
    for i from 0 to 6 while fini = false do
      if happy(\( b + \text{onestep}(c + i^2) \)) = -1 then fini := true; end if;
    end do;
    if fini = false then  # now we check for last digit 5
      for k from 0 to floor\( \left( \frac{b}{81} \right) \) do
        for d2 from 0 to 8 do
          fini2 := false;
          for i from 7 to 7 while fini2 = false do
            if happy(\( b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}( (c + i^2) \mod 100) \)) = -1 then fini2 := true; end if;
          end do;
          if fini2 = false then print(\( b, c, k, d2, \text{happy}(b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}( (c + 8^2) \mod 100) \)) end if;
        end do;
      end do;
    end if;
  end do;
end do;
```

```plaintext
> for b from 0 to 2000 do
  for c from 964 to 974 do  # so these have the last digit 5 carry but not any others.
    fini := false;
    for i from 0 to 5 while fini = false do
      if happy(\( b + \text{onestep}(c + i^2) \)) = -1 then fini := true; end if;
    end do;
    if fini = false then  # now we check for last digit 5
      for k from 0 to floor\( \left( \frac{b}{81} \right) \) do
        for d2 from 0 to 8 do
          end do;
        end do;
      end do;
    end if;
  end do;
end do;
```

\[ \text{(10)} \]
fini2 := false;
for i from 6 to 7 while fini2 = false do
  if happy(b + 2 \cdot d2 + 1 - k \cdot 81 + onestep\((c + i^2) \mod 100\)) = -1 then fini2 := true; end if;
end do;
if fini2 = false then print(b, c, k, d2, happy(b + 2 \cdot d2 + 1 - k \cdot 81 + onestep\((c + 8^2) \mod 100\)) ) end if;
end do;
end do;
for b from 0 to 2000 do # so these have the last digit 5 carry but not any others.
  fini := false;
  for i from 0 to 4 while fini = false do
    if happy(b + onestep(c + i^2)) = -1 then fini := true; end if;
  end do;
  if fini = false then # now we check for last digit 5
    for k from 0 to floor(b \cdot 81) do
      for d2 from 0 to 8 do # d2 is the digit of the b part that precedes the nines that carry.
        fini2 := false;
        for i from 5 to 7 while fini2 = false do
          if happy(b + 2 \cdot d2 + 1 - k \cdot 81 + onestep\((c + i^2) \mod 100\)) = -1 then fini2 := true; end if;
        end do;
        if fini2 = false then print(b, c, k, d2, happy(b + 2 \cdot d2 + 1 - k \cdot 81 + onestep\((c + 8^2) \mod 100\)) ) end if;
      end do;
    end do;
  end if;
end do;
end do;
end do;
end do;
end do:

1663, 978, 7, 4, 4

> for b from 0 to 2000 do # so these have the last digit 5 and 4 carry but not any others.
  fini := false;
  for i from 0 to 3 while fini = false do
    if happy(b + onestep(c + i^2)) = -1 then fini := true; end if;
  end do;
  if fini = false then # now we check for last digit 5, 4
    for k from 0 to floor(b \cdot 81) do
      for d2 from 0 to 8 do
        fini2 := false;
        for i from 4 to 7 while fini2 = false do
          if happy(b + 2 \cdot d2 + 1 - k \cdot 81 + onestep\((c + i^2) \mod 100\)) = -1 then fini2 := true; end if;
        end do;
      end do;
    end do;
  end if;
end do;
end do;
end do:

1663, 978, 7, 4, 4
for $b$ from 0 to 2000 do
for $c$ from 991 to 995 do  # so these have the last digit 5,4,3 carry but not any others.
  fini := false;
  for $i$ from 0 to 2 while fini = false do
    if $\text{happy}(b + \text{onestep}(c + i^2)) = -1$ then fini := true; end if;
  end do;
if fini = false then  # now we check for last digit 5,4,3
  for $k$ from 0 to floor($\frac{b}{81}$) do
    for $d2$ from 0 to 8 do
      fini2 := false;
      for $i$ from 3 to 7 while fini2 = false do
        if $\text{happy}(b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}( (c + i^2) \mod 100)) = -1$ then fini2 := true; end if;
      end do;
      if fini2 = false then  # now we check for last digit 5,4,3,2
        for $k$ from 0 to floor($\frac{b}{81}$) do
          for $d2$ from 0 to 8 do
            fini2 := false;
            for $i$ from 2 to 7 while fini2 = false do
              if $\text{happy}(b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}( (c + i^2) \mod 100)) = -1$ then fini2 := true; end if;
            end do;
            if fini2 = false then print($b$, $c$, $k$, $d2$, $\text{happy}(b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}( (c + 8^2) \mod 100))$) end if;
          end do;
        end do;
      end if;
    end do;
  end do;
end if;
end do;
end do;

end do;
if fini2 = false then print($b$, $c$, $k$, $d2$, $\text{happy}(b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}( (c + 8^2) \mod 100))$) end if;
end do;
end do;
end do;
end do:

1582, 994, 16, 7, -1

(12)
Clearly we need to investigate the \((b,c) = (1103, 934)\) case to see if we can find seven in a row before the carry.
We add nines to the "After the carry" minimal value to see if

\[
\text{minimalN}(1103); \quad 1799999999999999
\]

\[
\text{a} := 179999999999999934; \quad \text{a} := 179999999999999934
\]

We see here the best digit \(d2\) would be 8, but only by a couple digits, so we only need to extend the search a few values of \(k\) beyond the lowest \(k\) value that will give seven before the carry.

\[
\text{a} := 179999999999999934; \quad \text{for d2 from 0 to 8 do}
\]

\[
\text{print}(d2, \text{minimalbigN}(a - d2^2))
\]

\[
\text{end do;}
\]

\[
\begin{align*}
0, & \quad [1789999, 2222222222222216] \\
1, & \quad [789999, 2222222222222216] \\
2, & \quad [2599999, 2222222222222216] \\
3, & \quad [3788999, 2222222222222216] \\
4, & \quad [7799999, 2222222222222216] \\
5, & \quad [5899999, 2222222222222216] \\
6, & \quad [18888999, 2222222222222215]
\end{align*}
\]
\[ \begin{align*}
7, [1899999, 2222222222222215] \\
8, [1799999, 2222222222222215] 
\end{align*} \]

```plaintext
> a := 179999999999999934; # this is the value after the carry
ktop := iquo(a, 81);
for k from 0 to ktop do # k is the number of nines at the end of N1.
    for d2 from 0 to 8 do
        # d2 is the digit that precedes the k digits of nine before the carry happens, so it is d2 +1 after the carry.
        q := a - (d2 + 1)^2 + d2^2 + k*81;
        if happy(q + 9^2) > 0 and happy(q + 8^2) > 0 and happy(q + 7^2) > 0 and happy(q + 6^2) > 0
            and happy(q + 5^2) > 0 and happy(q + 4^2) > 0 and happy(q + 3^2) > 0
            then print(a, k, d2, happy(q + 2^2), happy(q + 1^2), minimalbigN(a - (d2 + 1)^2) end if;
        end do; end do:
```

```plaintext
if happy(q + 8^2) > 0 and happy(q + 7^2) > 0 and happy(q + 6^2) > 0
    then print(a, k, d2, happy(q + 2^2), happy(q + 1^2), minimalbigN(a - (d2 + 1)^2) end if;
```

```plaintext
a := 179999999999999934
ktop := 2222222222222221
179999999999999934, 97388, 3, -1, 7, [779999, 2222222222222216]
Warning, computation interrupted
```

<table>
<thead>
<tr>
<th>7, 1899999, 2222222222222215</th>
</tr>
</thead>
<tbody>
<tr>
<td>8, [1799999, 2222222222222215]</td>
</tr>
</tbody>
</table>

\[ a := 179999999999999934 \]

```plaintext
ktop := iquo(a, 81);
for k from 0 to ktop do # k is the number of nines at the end of N1.
    for d2 from 0 to 8 do
        # d2 is the digit that precedes the k digits of nine before the carry happens, so it is d2 +1 after the carry.
        q := a - (d2 + 1)^2 + d2^2 + k*81;
        if happy(q + 9^2) > 0 and happy(q + 8^2) > 0 and happy(q + 7^2) > 0 and happy(q + 6^2) > 0
            and happy(q + 5^2) > 0 and happy(q + 4^2) > 0 and happy(q + 3^2) > 0
            then print(a, k, d2, happy(q + 2^2), happy(q + 1^2), minimalbigN(a - (d2 + 1)^2) ) end if;
        end do; end do:
```

\[ a := 179999999999999934 \]

```plaintext
ktop := 2222222222222221
179999999999999934, 97388, 3, -1, 7, [779999, 2222222222222216]
```

Warning, computation interrupted

```plaintext
> k;
189637
```

```plaintext
> minimalbigN(179999999999999934 - 4^2); onestep(179999999999999934);
[779999, 2222222222222216]
```

```plaintext
1103
```

```plaintext
> for j from 0 to 8 do
    print(j, happy(onestep(77) + 2222222222222220*81 + 4^2 + j^2) ) end do;
0, 5
1, 8
2, 6
3, 5
4, 5
5, 5
6, 6
7, 6
8, -1
```

so we see that exactly 8 values after the carry work.

Next we will see that the seven before the carry work.

```plaintext
N = 77.(2222222222222220 nines).3.(97388 nines).3
```

```plaintext
> for j from 2 to 9 do
    print(j, happy(onestep(77) + 2222222222222220*81 + 3^2 + 97388*81 + j^2) ) end do;
2, -1
3, 7
4, 8
```

7, [1899999, 2222222222222215]
8, [1799999, 2222222222222215]
Referring back to the calculations done for the 14 in a row, we see that the $b=306$ and $c=355$ was the smallest possible case of 7 in a row before the carry, and we have this case here.

\[
5, 8 \\
6, 8 \\
7, 6 \\
8, 6 \\
9, 7
\]  
(19)

We next try for an 8/7 split over the carry, first looking for eight before the carry.

\[
> \ a1 := \text{onestep}(77) + 2222222222222220.81 + 3^2 + 97388.81; \\
\quad \ a1 := 180000000007888355
\]

(20)

\[
> \ b := \text{onestep}(180000000078888); \ c := 355; \\
\quad \ b := 306 \\
\quad \ c := 355
\]

(21)

> We next try for an 8/7 split over the carry, first looking for eight before the carry.

\[
> \ \text{for} \ b \ \text{from} \ 0 \ \text{to} \ 2000 \ \text{do} \\
\quad \ \text{for} \ c \ \text{from} \ 0 \ \text{to} \ 918 \ \text{do} \\
\quad \quad \ \text{fini} := \text{false}; \\
\quad \quad \ \text{for} \ i \ \text{from} \ 9 \ \text{to} \ 2 \ \text{by} \ -1 \ \text{while} \ \text{fini} = \text{false} \ \text{do} \\
\quad \quad \quad \ \text{if happy}(b + \text{onestep}(c + i^2)) = -1 \ \text{then fini} := \text{true}; \ \text{end if}; \\
\quad \quad \ \text{end do}; \\
\quad \quad \ \text{if fini} = \text{false} \ \text{then print}(b, c, \text{minimalN}(b), \text{happy}(b + \text{onestep}(c + 2^2))) \ \text{end if}; \\
\quad \ \text{end do}; \\
\quad \ \text{end do}; \\
\quad \ \text{print}(b, c, \text{minimalN}(b), \text{happy}(b + \text{onestep}(c + 2^2)), \text{happy}(b + \text{onestep}(c + 0^2))); \\
\quad b := 1710; \ c := 534; \\
\quad \text{happy}(b + \text{onestep}(c + 1^2)); \ \text{happy}(b + \text{onestep}(c + 0^2)); \\
\quad b := 1710 \\
\quad c := 534 \\
\quad 5 \\
\quad 4
\]

(22)

> Wow! We jump straight from seven in a row before the carry to all ten in a row before the carry, or equivalently, after the carry.

> We still need to check all the ones that have carries in the M1 level, where $c + d0^2 >= 1000$.

To analyze this, recall that $N = N1.d0$, $M1 = S(N1)$, and we set $M1=M2.c$ where $c$ is the last three digits.

If $c=d0^2>=1000$ then we need to consider how the carry affects the M2 portion. Let $M2 = M3.d2.99..99$ where the end of $M2$ has exactly k2 nines with the digit $d2 < 9$. Then $S(M2) = S(M3) + d2^2 + k2*9^2$, so $S(M3) = S(M2) - d2^2 - k2*81$. Now the $c+9^2$ will give a carry $10.e.f$ where $e$ and $f$ are the last two digits of $c+9^2$. Thus, after this carry, we have $M3.(d2+1).00...000.e.f so S(M3.(d2+1).00...000.e.f) = S(M3) + (d2+1)^2 + S(e.f) = S(M2)-d2^2 + (d2+1)^2 - k*81 + S(e.f) = b+2*d2+1-k*81+S(e.f)$. This is what we need to check to see if it is happy.
When $c > 918$ we need to worry about the carry from $d_0 = 9$, $c > 935$ we also need to worry about the carry with $d_0 = 8$, when $c > 950$ also worry about the carry when $d_0 = 7^2$, etc.

The $b$ below corresponds to the $S(M2)$ as above.

We check for eight or more in a row before the carry.

```
> for b from 0 to 2000 do
  for c from 919 to 935 do  # so these have the last digit 9 carry but not any others.
    fini := false;
    for i from 2 to 8 while fini = false do
      if happy(b + onestep(c + i^2)) = -1 then fini := true; end if;
    end do;
    if fini = false then # now we check for last digit 9
      for k from 0 to floor(b/81) do
        for d2 from 0 to 8 do
          if happy(b + 2d2 + 1 - k*81 + onestep((c + 9^2) mod 100)) > 0 then print(b, c, d2, 
            happy(b + onestep(c + i^2)), happy(b + onestep(c)), minimalN(b - d2^2 - k*81) 
            * 10^{k+4} + d2 * 10^{k+3} + 10^k + 3 - 1000 + c, evalf(log10(minimalN(b - d2^2 - k*81) 
            * 10^{k+4} + d2 * 10^{k+3} + 10^k + 3 - 1000 + c)) end if;
        end do;
      end do;
    end if;
  end do;
end do:
1654, 934, 0, 8, 5, 4, 288999999999999999999998934, 24.46089784
1654, 934, 2, 1, 5, 4, 17899999999999999999999999199934, 25.25285303
1654, 934, 2, 4, 5, 4, 17799999999999999999999999499934, 25.25042000
1654, 934, 2, 7, 5, 4, 68899999999999999999999999939934, 24.83815619
1654, 934, 3, 3, 5, 4, 59999999999999999999999999399934, 24.77815125
1654, 934, 3, 5, 5, 4, 39999999999999999999999999999934, 24.60205999
1654, 934, 4, 8, 5, 4, 288999999999999999999999999899934, 24.46089784
1654, 934, 5, 6, 5, 4, 78899999999999999999999999999934, 24.9702196
1654, 934, 5, 7, 5, 4, 68899999999999999999999999999934, 24.83815619
1654, 934, 6, 7, 5, 4, 68899999999999999999999999999934, 24.83815619
1654, 934, 6, 8, 5, 4, 28899999999999999999999999999934, 24.46089784
1654, 934, 7, 0, 5, 4, 28899999999999999999999999999934, 25.46074754
1654, 934, 7, 4, 5, 4, 177999999999999999999999994999934, 25.25042000
1654, 934, 7, 7, 5, 4, 68899999999999999999999999999934, 24.83815619
1654, 934, 8, 0, 5, 4, 28899999999999999999999999999934, 25.46074754
1654, 934, 8, 3, 5, 4, 59999999999999999999999999999934, 24.77815125
1654, 934, 9, 6, 5, 4, 78899999999999999999999999999934, 24.89702196
1654, 934, 10, 5, 5, 4, 39999999999999999999999999999934, 24.60205999
1654, 934, 11, 0, 5, 4, 28899999999999999999999999999934, 25.46074754
1654, 934, 11, 6, 5, 4, 7889999999699999999999999999934, 24.89702196
1654, 934, 11, 8, 5, 4, 2889999998999999999999999999934, 24.46089784
for $b$ from 0 to 2000 do
  for $c$ from 936 to 950 do  # so these have the last digit 9 and 8 carry but not any others.
    fini := false;
    for $i$ from 2 to 7 while fini = false do
      if happy($b + \text{onestep}(c + i^2)$) = -1 then fini := true; end if;
    end do;
    if fini = false then  # now we check for last digit 9 and 8
      for $k$ from 0 to floor($\frac{b}{81}$) do
        for $d2$ from 0 to 8 do
          fini2 := false;
          for $i$ from 8 to 9 while fini2 = false do
            if happy($b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}( (c + i^2) \mod 100 ) ) = -1 then fini2 := true; end if;
          end do;
          if fini2 = false then print $b$, $c$, $k$, $d2$, happy($b + \text{onestep}(c + i^2)$), happy($b + \text{onestep}(c)$),
            \text{minimalN}( b - d2^2 - k \cdot 81 ) \cdot 10^{(k+4)} + d2 \cdot 10^{k+3} + 10^{k+3} - 1000 + c,
            eva[	ext{lf}(\log10(\text{minimalN}( b - d2^2 - k \cdot 81 ) \cdot 10^{(k+4)} + d2 \cdot 10^{k+3} + 10^{k+3} - 1000 + c ) ))
          end if;
        end do;
      end if;
    end do;
  end do;
end do;

for $b$ from 0 to 2000 do
  for $c$ from 951 to 963 do  # so these have the last digit 9,8,7 carry but not any others.
    fini := false;
    for $i$ from 2 to 6 while fini = false do
      if happy($b + \text{onestep}(c + i^2)$) = -1 then fini := true; end if;
    end do;
    if fini = false then  # now we check for last digit 9,8,7
      print $b$, $c$;
    end do;
end do;

(24)
for $k$ from 0 to $\text{floor} \left( \frac{b}{81} \right)$ do

for $d2$ from 0 to 8 do
  fini2 := false;
  for $i$ from 7 to 9 while fini2 = false do
    if $\text{happy}(b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}(\ (c + i^2) \mod 100 )) = -1$ then fini2 := true; end
    if;
  end do;
  if fini2 = false then print($b, c, k, d2, \text{happy}(b + \text{onestep}(c + i^2)),$ happy($b + \text{onestep}(c)),$
    $\text{minimalN}(b - d2^2 - k \cdot 81) \cdot 10^{(k+4)} + d2 \cdot 10^{k+3} + 10^k - 1000 + c,$
    $\text{evalf}(\log10(\text{minimalN}(b - d2^2 - k \cdot 81) \cdot 10^{(k+4)} + d2 \cdot 10^{k+3} + 10^k - 1000 + c)))$ end if;
  end do;
end if;
end do;
end do;

> for $b$ from 0 to 2000 do
  for $c$ from 964 to 974 do # so these have the last digit 9,8,7,6 carry but not any others.
    fini := false;
    for $i$ from 2 to 5 while fini = false do
      if $\text{happy}(b + \text{onestep}(c + i^2)) = -1$ then fini := true; end if;
    end do;
    if fini = false then # now we check for last digit 9,8,7,6
      for $k$ from 0 to $\text{floor} \left( \frac{b}{81} \right)$ do
        for $d2$ from 0 to 8 do
          fini2 := false;
          for $i$ from 6 to 9 while fini2 = false do
            if $\text{happy}(b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}(\ (c + i^2) \mod 100 )) = -1$ then fini2 := true; end if;
          end do;
          if fini2 = false then print($b, c, k, d2, \text{happy}(b + \text{onestep}(c + i^2)),$
            $\text{minimalN}(b - d2^2 - k \cdot 81) \cdot 10^{(k+4)} + d2 \cdot 10^{k+3} + 10^k - 1000 + c,$
            $\text{evalf}(\log10(\text{minimalN}(b - d2^2 - k \cdot 81) \cdot 10^{(k+4)} + d2 \cdot 10^{k+3} + 10^k - 1000 + c)))$ end if;
          end do;
        end if;
      end do;
  end do;
end do;

> for $b$ from 0 to 2000 do
  for $c$ from 975 to 983 do # so these have the last digit 9,8,7,6,5 carry but not any others.
    fini := false;
    for $i$ from 2 to 4 while fini = false do
      if $\text{happy}(b + \text{onestep}(c + i^2)) = -1$ then fini := true; end if;
    end do;
    if fini = false then # now we check for last digit 9,8,7,6,5
      for $k$ from 0 to $\text{floor} \left( \frac{b}{81} \right)$ do
        for $d2$ from 0 to 8 do
          fini2 := false;
          for $i$ from 6 to 9 while fini2 = false do
            if $\text{happy}(b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}(\ (c + i^2) \mod 100 )) = -1$ then fini2 := true; end if;
          end do;
          if fini2 = false then print($b, c, k, d2, \text{happy}(b + \text{onestep}(c + i^2)),$
            $\text{minimalN}(b - d2^2 - k \cdot 81) \cdot 10^{(k+4)} + d2 \cdot 10^{k+3} + 10^k - 1000 + c,$
            $\text{evalf}(\log10(\text{minimalN}(b - d2^2 - k \cdot 81) \cdot 10^{(k+4)} + d2 \cdot 10^{k+3} + 10^k - 1000 + c)))$ end if;
          end do;
        end if;
      end do;
    end do;
end do;
for \( k \) from 0 to floor \( \left( \frac{b}{81} \right) \) do

for \( d2 \) from 0 to 8 do

fini2 := false;

for \( i \) from 5 to 9 while fini2 = false do

if \( \text{happy}(b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}(c + i^2 \mod 100)) = -1 \) then fini2 := true; end if;
end do;

if fini2 = false then print(b, c, k, d2, \( \text{happy}(b + \text{onestep}(c + i^2)) \), \( \text{happy}(b + \text{onestep}(c)) \), \( \text{minimalN}(b - d2^2 - k \cdot 81) \cdot 10^{(k+4)} + d2 \cdot 10^k + 3 \cdot 10^{k+3} - 1000 + c \), \( \text{evalf}(\log10(\text{minimalN}(b - d2^2 - k \cdot 81) \cdot 10^{(k+4)} + d2 \cdot 10^k + 3 \cdot 10^{k+3} - 1000 + c)) \) ) end if;
end do;
end if;
end do;
end do;
end do;
end if;
end do:

1647, 978, 6, 3, -1, 5, 17799999999999993999999978, 25.25042000
1654, 980, 17, 8, -1, 1, 28898999999999999999999980, 24.46088281
1663, 978, 7, 4, 7, 56889999999999999999999978, 25.75503593

> for \( b \) from 0 to 2000 do

for \( c \) from 984 to 990 do # so these have the last digit 9,8,7,6,5,4 carry but not any others.

fini := false;

for \( i \) from 2 to 3 while fini = false do

if \( \text{happy}(b + \text{onestep}(c + i^2)) = -1 \) then fini := true; end if;
end do;

if fini = false then # now we check for last digit 9,8,7,6,5,4

for \( k \) from 0 to floor \( \left( \frac{b}{81} \right) \) do

for \( d2 \) from 0 to 8 do

fini2 := false;

for \( i \) from 4 to 9 while fini2 = false do

if \( \text{happy}(b + 2 \cdot d2 + 1 - k \cdot 81 + \text{onestep}(c + i^2 \mod 100)) = -1 \) then fini2 := true; end if;
end do;

if fini2 = false then print(b, c, k, d2, \( \text{happy}(b + \text{onestep}(c + i^2)) \), \( \text{happy}(b + \text{onestep}(c)) \), \( \text{minimalN}(b - d2^2 - k \cdot 81) \cdot 10^{(k+4)} + d2 \cdot 10^k + 3 \cdot 10^{k+3} - 1000 + c \), \( \text{evalf}(\log10(\text{minimalN}(b - d2^2 - k \cdot 81) \cdot 10^{(k+4)} + d2 \cdot 10^k + 3 \cdot 10^{k+3} - 1000 + c)) \) ) end if;
end do;
end if;
end do;
end do:

1647, 978, 6, 3, -1, 5, 17799999999999993999999978, 25.25042000
1654, 980, 17, 8, -1, 1, 28898999999999999999999980, 24.46088281
1663, 978, 7, 4, 7, 56889999999999999999999978, 25.75503593

> for \( b \) from 0 to 2000 do

for \( c \) from 991 to 995 do # so these have the last digit 9,8,7,6,5,4,3 carry but not any others.

fini := false;

for \( i \) from 2 to 2 while fini = false do
So the candidate values of $b$ are 1710, 1654, 1602, 1647, 1654, and 1663. The smallest $M_1$ value resulting from these is the 25 digit number

$$24.60205999$$
M1 = 1888899999999999799999949

corresponding to \((b, c, k, d2) = (1602, 949, 5, 7)\) which has precisely eight consecutive happy numbers
before the carry.

This M1 much larger than the eighteen digit M1 value we got from the 7/8 split over the carry, hence
the 8/7 split over the carry cannot give a smaller N.

Thus, the 18 digit number above must yield the best value for N.