CRACK A HACK

Champernowne’s constant

Course: ALGORITHMIC PROBLEM SOLVING
Course code: 17ECSE309

By: Ananth Bhattramakki
USN: 01fe15bec025

1. Introduction

In mathematics, the **Champernowne constant** C_{10} is a transcendental real constant whose decimal expansion has important properties. It is named after economist and mathematician D. G. Champernowne, who published it as an undergraduate in 1933.[1]

For base 10, the number is defined by concatenating representations of successive integers:

C_{10} = 0.12345678910111213141516…

Champernowne constants can also be constructed in other bases, similarly, for example:

C_2 = 0.11011100101110111…
C_3 = 0.12101112202122….

The Champernowne constants can be expressed exactly as infinite series:

$$C_m = \sum_{n=1}^{\infty} \frac{n}{10_b^{\left(\sum_{k=1}^{n}\left\lceil \log_{10_b}(k+1)\right\rceil\right)}}$$

2. Logic

Champernowne's constant can be looked at as a concatenation of many series of concatenated counting numbers. Each series represents an order of magnitude or, more succinctly, a power of 10 as follows:

Series (*k*)	**Range**	**Number of terms**	**Number of characters in series**	**Number of characters total**
1	1-9	9	9	9
2	10-99	90	180	189
3	100-999	900	2700	2889
4	1000-9999	9000	36000	38889
5	10000-99999	90000	450000	488889

We can make a few observations:

- The size of each term in the series is the series number, *k*. That is, series 4 is composed of 4-digit numbers.
- The range for each series, *k*, is $10k^{-1}$ to $10k$-1 and has $9 \cdot 10k^{-1}$ terms.
- The number of digits in each series is *k* * number of terms. So, series 4 is 4 * 9000 = 36000 digits

3. Algorithm

The steps would be:

- Check where the required digit **‘g’** lies in the number of characters. This gives the value of ‘**k**’ which is the number of digits in the series.
- Now subtract **g** with the number of characters in previous series. So that the required term is that much far from the previous series.
- Now apply **g :=(g-1)/k** to get the original placement of the digit from the previous series. And the remainder will give the exact digit of the number we extract.
- Now **g:=g+(“Starting element of this series”)** will give the number we require. Using the remainder from the previous step as the index of the extracted number , we get the required digit.

For Example:

The constant is: 0.12345678910111213141516171819202122…99
Series 1: 1,2,3,4,5,6,7,8,9
Series 2: 10,11,12,13,14,15,16,17,1**8**,19,20,21,22,…,99

- All indexes, except the series, start with a base of zero. The 18th term is index 17; always one less.
- We would first determine the 27th digit is in the 2nd series as $9 < 27 \leq 9+180$.
- Since all terms in the 2nd series are 2-digits we find our term index 27-9 = 18th term in the constant and (18-1)/2 = 8 (remainder 1) as the series index or 9th in the series.
- Adding 10 (the start of the 2nd series) to 8 gives us our value as 18.
- The remainder is our last index and, again, starting from 0, points to the digit in the term. In this example the 2nd (index 1) character of 18 is the number 8.

The 27th digit in the constant is 8.

4. Code

```
def champ(f):                              //calculate product di1*di2*di3….*di7
  product =1
  for g in f:
    product*=calculate(g)
  print product

def calculate(g):
  global base                              //list with the number of total characters for corresponding k

  for i in range(len(base)):               //finding where ‘g’ lies
    if base[i]<g<=base[i+1]:
      index=i+1                            // k
      g-=base[i]                           //distance from previous series

  g,remainder=divmod(g-1,index)
  g+=pow(10,(index-1))                     //The number required
  return int(str(g)[remainder])            // the digit of the required number
```

```
base=[0,9,189,2889,38889,488889,5888889,68888889,788888889,8888888889,98888888889,10888
88888889,11888888888889,128888888888889,1388888888888889,14888888888888889,15888888
8888888889,1688888888888888889,17888888888888888889]

                                             //base list calculated from formula 9*k*10^(k-1)
                                             //upto k=18

t=int(raw_input())
for i in range(t):
   f=list(map(int,raw_input().strip().split(' ')))      //input list
   champ(f)
```

5. References

1).https://en.wikipedia.org/wiki/Champernowne_constant
2). http://mathworld.wolfram.com/ChampernowneConstantDigits.html
3) https://blog.dreamshire.com
4). Champernowne, D. G. "The Construction of Decimals Normal in the Scale of Ten." *J. London Math. Soc.* **8**, 1933