Code Jam 2009 - Round 1C

Analysis: All Your Base

```
IN A.D. 2101
WAR WAS BEGINNING
...
CATS: HOW ARE YOU GENTLEMEN !!
CATS: ALL YOUR BASE ARE BELONG TO US
```

This problem, of course, takes place in the halcyon days of A.D. 2100, before all our base became belong to Cats. It also firmly demonstrates that its author still lives in A.D. 2001.

In this problem our job is to read a series of symbols and interpret them as digits of a number in some base. In order to *minimize* the number, we'll want to use the smallest base possible. We can check how many different characters show up in the number; if that number of characters is **k**, then we can work in base `max(k,2)` (base 1 is not allowed). From there, it's simply a matter of assigning values from 0 to **k** to all of the characters. We can't start with a 0, so we should make the first digit a 1; and after that, simply assign the lowest number available to characters from left to right. In that way, "ab2ac999" becomes "10213444" in base 5, or 85499 seconds.

The following is a brief solution in Python:

```
import sys

N = int(sys.stdin.readline().strip())
for qw in range(1, N+1):
  print 'Case #%d:' % qw,

  num = sys.stdin.readline().strip()
  values = {num[0]: 1}
  for c in num:
    if c not in values:
      sz = len(values)
      if sz == 1:
        values[c] = 0
      else:
        values[c] = sz
  result = 0
  base = max(len(values), 2)
  for c in num:
    result *= base
    result += values[c]
  print result
```

One final note: on our official Google Group, user Damien pointed out the following:

"The first example case in All Your Base implies the alien language uses a left-right notation. If that assumption is wrong and they actually used right-left you'd be 54 seconds late for the war: 11001001 binary = 201 decimal, but the reverse 10010011 is 147 decimal. Dangerous assumption to make don't you think? :-)"

I hate to say it, but he's right; even though 2219 people solved this problem, we may still be taken by surprise!