Life Forms

* How to run
  + In UVa: C++11 5.3.0 - GNU C++ Compiler with options: -lm -lcrypt -O2 -std=c++11 -pipe -DONLINE\_JUDGE
  + Local machine: g++ -std=c++11 SuffixArray.cpp
* Idea
  + In one scenario, connect all forms into one string/char array
    - Separate these forms by using character that is not appeared in the forms
    - E.g. “abcdefg” and “bcdefgh” ⇒ abcdefg.bcdefgh
  + Use Suffix Array algorithm to get suffix array (sa)
  + Get height array by using suffix array, according to The Kasai et al. algorithm (<http://www2.cs.sfu.ca/~binay/2018/409/suffix-array.pdf>)
  + Use binary search to get the right longest common substring length
    - The condition within binary search is that if the height array has the height that appeared one than half of n, then it is valid. Otherwise, it is invalid.
  + Print results by requirements.
* Data Structure and Algorithm
  + Suffix array
    - Modified from <https://github.com/jaehyunp/stanfordacm/blob/master/code/SuffixArray.cc>
  + The Kasai et al. algorithm
    - <https://www.hackerrank.com/topics/lcp-array>
    - <https://pdfs.semanticscholar.org/f5e5/f365acc6f00c014c523c65efd9df6cee2606.pdf>
    - <http://www2.cs.sfu.ca/~binay/2018/409/suffix-array.pdf>
  + Binary search