CIS 350/1	Program 1	Winter 2015

Due: January 27th
Purpose : Learn about the STL, get feet wet, test cases, learn about the autograder

Nick Cole, Cyber Attorney, has a client up on a murder charge. He needs your help to convince a jury that his client is innocent. You will do this by writing a computer program that will help compute number of full strands of DNA which could possibly match with the collected DNA .
At the crime scene, CSI (Crime Scene Investigations) unit has collected traces of DNA. Recall that DNA sequences can be represented as strings of characters on the four letter alphabet (a,c,t, g) where letter represents a nucleotide. Once CSI has collected the DNA they try to match this to your client using a technique called sequencing by hybridization (SBH). The technique separates the collected DNA into all of its possible substrings of length *k*. Once these are identified, then all possible strands with only the given substrings are identified. The number of all possible strands of a given length with the specified substrings divided by the total number of possible strands of that length gives an estimate to the probability that an arbitrary persons DNA will match with the crime scene DNA.
For example, suppose we know that ac, ca, and cc are the only length-2 substrings of strand S. It is certainly possible that acca is a substring of S, since the center (cc) is one of our possibilities. However, caac *cannot* be a substring of S, since we know that aa is not a substring of S. We needed to find a fast algorithm to construct all the consistent length-2k strings, since S could be very long.

The simplest algorithm to build the 2k strings would be to concatenate all pairs of k-strings together, and then for each pair to make sure that all (*k*-1) length-*k* substrings spanning the boundary of the concatenation were in fact substrings.

For example if the 2 length 5 strings are acgtt and atcca then the middle strings to consider are :

```
      t a t c c
    t t a t c
  g t t a t
c g t t a
a c g t t a t c c a
```

Consider the nine possible concatenations of ac, ca and cc are acac, acca, accc, caac, caca, cacc, ccac, ccca, and cccc. Only caac can be eliminated because of the absence of the substring aa.
Consider if the three strands acc, cac, and cca are collected then the possible stands of length 6 are :
accacc,~~aeeeea~~,~~aeeeae~~,~~eeaaee~~,ccacca,~~eeaeae~~,~~eaeaee~~,~~eaeeea~~,caccac.
While the stands of length 12 are :
accaccaccacc, accaccccacca, ccaccacaccac, ccaccaccacca

Input
The input will consist of multiple problem instances. The first line will consist of an integer *p* indicating the number of problem instances. Each instance will begin with 1 integer (*n*) indicating the number of substrings of same length that we know are substrings of S and a target length *t*. The target length *t* will always be a power of 2 times the length of the input. The next line will contain the *n* substrings of length *k* (separated by spaces) that we know are the only substrings of length *k* that are in S

Output
Output for each problem instance will consist of 1 integer which contains the number of strings of length *t* that we know are in S.

Sample Input
```
4
3 4
ac ca cc
3 6
acc cgt act
3 6
acc cca cac
3 12
acc cca cac
```

Sample Output
```
8
0
3
3
```

REQUIRED ALGORITHM
1. make all the length 2*k* strings
   a. For each of the length 2*k* strings find all *k*-1 substrings of length *k* that straddle the middle.
   b. Keep each of these strings in a vector
   c. Go back to 1 to make strings twice as long until the strings in the vector are of the appropriate length.
2. Return the number of strings in your vector.

How the program will be graded

What _ indicates program/other is memo	Points	Due Jan 27th
External Documentation	**1**	
Your Name	1	
Analysis	**26**	
Set of 4 Test Cases (reason for test, test input and expected outputs, actual output)	10	
Time Analysis Worst case for O(based on the number of strands and the length of the strands and the target length) of each function with explanation. (main is a function)	8	
Space Analysis Worst case O(based on the number of strands and the length of the strands and the target length) of each function with explanation. (main is still a function)	8	
Program Listing Style	**23**	
Your Name	1	
Description of the problem	2	
Style (variable names, length of functions, lack of global variables, independence of functions)	10	
Pre/Post conditions for functions	10	
Functionality	**50**	
Main and rest	8	
Reading the input (a function) (series of calls to possibly unwritten functions)	5	
Echo input to screen (as debug) Commented out in final version	5	
Computing the number of possible strands (based on the autograder)	32	

.