CS007A: Word Squares

A word square is an arrangement of letters in a square such that each row forms a word from left to right and each column forms a word from top to bottom.

AAH APE SET	AAH APE SEW	AAH APE SEX	AAH APE SON	AAH APE SOP	AAH APE SOT	AAH APE SOW	AAH APE SOX	AAH APE SOY	AAH APE SPY
AAH APE STY	AAH APO STY	AAH ARE HEY	AAH ARE HMM	AAH ARE SEX	AAH GIE ART	AAH GNU ATE	AAH GNU EAT	AAH GNU EYE	AAH GNU OAT
AAH GNU ODE	AAH GNU SAG	AAH GNU SIN	AAL CRY EKE	AAL DAY SHE	AAL DIE DRY	AAL DIE ODA	AAL DIE ODD	AAL DIE ODE	AAL DIE OLD
AAL DIE ONE	AAL DIE SLY	AAL DIE STY	AAL DUE SKA	AAL DUE SKY	AAL DYE ZED	AAL FIE TRY	AAS WHA LIP	AAS WHA LIT	AAS WHO LIB
ABO IRK TOE	ABS ROT TOY	ABS SIP SOY	ABS SUP STY	ABS ZOO OWL	ABY BYE YEW	ABY DYE SEW	ABY HUE SYN	ACT RUE TEN	ACT SHY PIE
ACT TRY EYE	ACT YOW EGO	ADD AIR SPY	ADD BYE YEN	ADO AIR SET	ADO BAD SPA	ADO GUN SHE	ADO NOD YES	ADS CUP TOY	ADS DOT SLY
ADS DYE SEX	ADS FOE TET	ADS GAY OWN	ADS MAT PLY	ADS NUT TOY	ADS POL TRY	ADS RIP TEA	ADS SUP SHY	ADS WAR AHI	ADS YAK SPY

The object of this project is to investigate algorithms for developing these using a dictionary of words.

You will ultimately want a good list of words. You can obtain dictionaries of three letter words and four letter words. Three letter words and their definitions can be found here:
http://www.yak.net/kablooey/scrabble/3letterwords.html
You can write a script using the `getline()` function to strip the words away from their definitions into another file, though there are other sources that would be easier to copy/paste into a .txt file, say here: http://www.scrabble-word-finder.com/word-list/csw07/3-letter-words.html

Here is some code we developed in class to produce 3by3 word squares such as those above:

As you can see, there are just two functions, `printArray()` and `checkCatDog()`. A large part of the work, getting a word square from the dictionary, is left to `main()` – this is more properly pulled out as another function, say, `getCatDog()`.

Your task: Based on our discussion in class as to how this program works, modify it to work on a dictionary of 4-letter words. Note that a good way to test the program is to use an artificial list of words like, “ABAB, DADA, BABA, ADAD”. As it is, the three-letter word square program shown here, as applied to a robust list of about 1000 three-letter-words, make take more than 24 hours to complete. Investigate other ways of speeding up the algorithm.

```
// Produce and check 3x3 letter word squares.

#include <iostream>
#include <fstream>

using namespace std;

void printArray(char [][3]); // display 3by3 word square
bool checkCatDog(char [][3]); // check to see if the columns also are words


int main()
{
	// Each temp string is read into catdog[3][3]
	char tmpStr1[4], tmpStr2[4], tmpStr3[4];
	// A contender to be checked
	char catdog[3][3];
	// read through the same file for each row with different ifstreams
	ifstream infile1, infile2, infile3;
	infile1.open("3letWrdsNoDef.txt");
	infile2.open("3letWrdsNoDef.txt");
	infile3.open("3letWrdsNoDef.txt");
	while(infile1 >> tmpStr1)
	{
		for(int j = 0; j < 3; ++j) // start loading up the catdog square
			catdog[0][j] = tmpStr1[j];
		while(infile2 >> tmpStr2)
		{
			for(int j = 0; j < 3; ++j)
				catdog[1][j] = tmpStr2[j];
			while(infile3 >> tmpStr3)
			{
				for(int j = 0; j < 3; ++j)
					catdog[2][j] = tmpStr3[j];
				// checkCatDog() returns true iff the
				// columns also from words
				if(checkCatDog(catdog)) printArray(catdog);
			}
			infile3.clear();
			infile3.seekg(0L,ios::beg);
		}
		// clear the end of file flag as each ifstream reaches the end
		infile2.clear();
		// …and reset it to the beginning again
		infile2.seekg(0L,ios::beg);
	}
	return 0;
}

void printArray(char s[][3])
{
	for(int i = 0; i < 3; ++i)
	{
		for(int j = 0; j < 3; ++j)
		    cout << s[i][j];
		cout << endl;
	}
	cout << endl;
}
```

```
bool checkCatDog(char s[][3])
{
	bool wd1 = false, wd2 = false, wd3 = false;
	char wrd1[4], wrd2[4], wrd3[4], word[4];
	for(int i = 0; i < 3; i++)
	{
		wrd1[i] = s[i][0];
		wrd2[i] = s[i][1];
		wrd3[i] = s[i][2];
	}
	//wrd1[4] = '\0'; wrd2[4] = '\0'; wrd3[4] = '\0';
	ifstream infile;
	infile.open("3letWrdsNoDef.txt");
	while(infile >> word)
	{
		// This is awkward - improve on it.
		if(word[0] == wrd1[0] &&
		   word[1] == wrd1[1] &&
		   word[2] == wrd1[2]) wd1 = true;
		if(word[0] == wrd2[0] &&
		   word[1] == wrd2[1] &&
		   word[2] == wrd2[2]) wd2 = true;
		if(word[0] == wrd3[0] &&
		   word[1] == wrd3[1] &&
		   word[2] == wrd3[2]) wd3 = true;
	}
	return bool(wd1 && wd2 && wd3);
}
```