```
//Problem 1

function countPairs (String sentence) {
        collection of WORDS;
        stringstream ss;

        while (stringstream !failing) {
                String current_word;
                ss << sentence;
                ss >> current_word;

                add current_word to WORDS
        }

        for (every word in WORDS) {
                for(every word between current word and last word) {
                        if(pair(current_word, next_word) does not exist in PAIRCOUNTER){
                                add (current_word, next_word) with int 1 to PAIRCOUNTER;
                        }

                        else if (pair does exist in PAIRCOUNTER) {
                                add 1 to its int in PAIRCOUNTER;
                        }

                }

        }
}

function findMostFreqPair() {
        mostFreqCount = 0;
        for (every element in PAIRCOUNTER) {
                if (element's int > mostFreqCount) {
                        mostFreqWordPair = element's pair of strings;
                        mostFreqCount = element's int;
                }
        }
}
```

```
main {
    //this will keep count of the occurences of word-pairs.
    map of <pair of strings, int> PAIRCOUNTER;
    String mostFreqWordPair;
    int mostFreqCount;

    for (every sentence in sentenceSplitter(article)) {
        countPairs(sentence);
    }

    findMostFreqPair();

    print "Most frequent word pair is : " + mostFreqWordPair  + "\n";
    print "Frequency : " + mostFreqCount;

}
```