```
//declare a struct or class that defines what words will be paired with the other word then places a count along with it.

Struct pair

        Word 1

        Word 2

        Frequency

// take the input and format it with a vector made with the constructed Struct

Vec<pair> deuce

//long string for input

String input = Get line (string, in)

//loop back if empty()

If (input. Size() == null)

String input = Get line (string, in)

//output in another string with frequency

//using the given sentence splitter function break the input into sentences.

//for each sentence in the defined input tokenize it by breaking down the sentence's further into words.

//using the newly formed individual words; for each word grab two unequal words and group them together in pairs

//not adding duplicates

// assign the newly grouped words into the declared Struct

//for each word pair in deuce compare with the word pair in Struct pair

//if pair exists frequency ++

//if else then the pair is not there and needs to be pushed_back into newly formed pair

//new method to find the highest frequency of word pairs

Void highest frequency of word pairs(deuce) {

//for each word pair in pair

//if frequency is > frequency in word pair

//new max frequency[location]

}
```

//Else if they have the same amount then do nothing

Return 0