```
getProductFrequency(Product current_product) {
        int tCount = 0;
        int finalFreq;

        for(every transaction in D) {
                if(current_product is in transaction) {
                        tCount++;
                }
        }

        finalFreq = (tCount / D.size()) * 100;

        if(finalFreq >= min_freq) {
                add current_product to FREQUENTPRODUCTS;
        }
}

getPairFrequency(Product current_product) {

        for(every product in FREQUENTPRODUCTS) {

                int tCount = 0;
                int finalFreq = 0;

                for(every transaction in D) {
                        if(current_product and product are in transaction) {
                                tCount++;
                        }
                }

                finalFreq = (tCount / D.size()) * 100;

                if(finalFreq >= min_freq) {
                        add pair (current_product, product) to FREQUENTPAIRS;
                }

        }
}
```

```
printFreqSet(set s) {
	if (s is empty) {

		print "none";

	} else {

		for (every element in s) {
			print element + "\n";
		}

	}
}


main {
	print "Enter minimum frequency : ";
	user input >> min_freq;

	set of FREQUENTPRODUCTS;
	set of (pair of <Product>) FREQUENTPAIRS;

	for(every Product in P) {
		getProductFrequency(product);
	}

	for(every Product in FREQUENTPRODUCTS) {
		getPairFrequency(product);
	}

	print "\nFrequent product(s) : \n"
	printFreqSet(FREQUENTPRODUCTS);

	print "\nFrequent product pair(s) : \n"
	printFreqSet(FREQUENTPAIRS);

}
```