Employee badge access - frequency

We want to find employees who badged into our secured room unusually often. We have an unordered list of names and access times over a single day. Access times are given as three or four-digit numbers using 24-hour time, such as "800" or "2250".

Write a function that finds anyone who badged into the room 3 or more times in a 1-hour period, and returns each time that they badged in during that period. (If there are multiple 1-hour periods where this was true, just return the first one.)

Expected output:

John: 830 835 855 915 930

Paul: 1315 1355 1405

Solution:

```
import java.io.*;

import java.util.*;

class Solution {

public static void main(String[] args) {

        String badgeRecords[][] = new String[][] { { "Martha", "exit" },
                                { "Paul", "enter" }, { "Martha", "enter" },
                                { "Martha", "exit" }, { "Jennifer", "enter" },
                                { "Paul", "enter" }, { "Curtis", "enter" },
                                { "Paul", "exit" }, { "Martha", "enter" },
                                { "Martha", "exit" }, { "Jennifer", "exit" } };

        String[][] badgeRecords2 = new String[][] { { "Paul", "1355" },
                                { "Jennifer", "1910" }, { "John", "830" },
                                { "Paul", "1315" }, { "John", "835" },
                                { "Paul", "1405" }, { "Paul", "1630" },
                                { "John", "855” }, { "John", "930" },
                                { "John", "915" }, { "Jennifer", "1335" },
                                { "Jennifer", "730" }, { "John", "1630" }, };

        List<String> res2 = new ArrayList<>();
        res2 = getFreqEmp(badgeRecords2);
        for(String s : res2){
                System.out.println(s);
        }

}

public static List<String> getFreqEmp(String badgeRecords[][]) {
        List<String> op = new ArrayList<>();
        Arrays.sort(badgeRecords, (String[] a1, String[] b1) -> Integer.parseInt(a1[1]) -
Integer.parseInt(b1[1]));

        for (int i = 0; i < badgeRecords.length; i++) {
```

```
                int curr = Integer.parseInt(badgeRecords[i][1]);
                int rem = curr % 100;
                int window = (((curr / 100) + 1) * 100) + (rem);
                int j = i + 1;
                while (j < badgeRecords.length && Integer.parseInt(badgeRecords[j][1]) <=
window)
                        j++;

                        HashMap<String, Integer> hm = new HashMap<>();
                        for (int k = i; k < j; k++) {
                                if (hm.containsKey(badgeRecords[k][0])) {
                                        int count = hm.get(badgeRecords[k][0]);
                                        hm.replace(badgeRecords[k][0], count + 1);
                                } else {
                                        hm.put(badgeRecords[k][0], 1);
                                }
                        }
                        for (String name : hm.keySet()) {
                                if (hm.get(name) >= 3 && (!op.contains(name)))
                                        op.add(name);
                        }

                }
                return op;
        }


}
```