

Q#時刻から最大時刻を求める

1日の時刻を「0時0分」から「23時59分」までの1分刻みで表現することを考える。

例えば「10時2分」は「1」「0」「2」の数字パーツから構成されている。ここで数字パーツを組み替えて新たな時刻を構成することを考える。すると「1時20分」「2時10分」そして元の時刻と同じ「10時2分」の3つの時刻を作ることができる。これらの時刻のうち、もっとも遅い時刻である「10時2分」を最大時刻と呼ぶことにする。「10時2分」の最大時刻は「10時2分」である。

同様に、「10時12分」は「1」「0」「1」「2」の数字パーツから構成されているので、最大時刻は「21時10分」である。「10時12分」の最大時刻は「21時10分」である。

1日の時刻		数字パーツ		最大時刻	
0 時 0 分	⇒	0 0	⇒	0 時 0 分	
0 時 1 分	⇒	0 1	⇒	1 時 0 分	※ 10 時 0 分 ではない。
:				:	
0 時 23 分	⇒	0 2 3	⇒	23 時 0 分	※ 3 時 2 分 ではない。
0 時 24 分	⇒	0 2 4	⇒	4 時 20 分	※ 24 時 0 分 ではない。
:				:	
1 時 0 分	⇒	1 0	⇒	1 時 0 分	
1 時 1 分	⇒	1 1	⇒	1 時 1 分	※ 11 時 0 分 ではない。
:				:	
13 時 40 分	⇒	1 3 4 0	⇒	14 時 30 分	
13 時 41 分	⇒	1 3 4 1	⇒	14 時 31 分	
13 時 42 分	⇒	1 3 4 2	⇒	23 時 41 分	※ 24 時 31 分 ではない。
:				:	
23 時 59 分	⇒	2 3 5 9	⇒	23 時 59 分	

問題1

指定された時刻（h時m分）から数字パーツ（n1, n2, ...）を求める関数を作成しなさい。たとえば h=10, m=2 が与えられたら [0, 1, 2] という配列を返さなければならない。

```
static int[] getNumbers(int h, int m) { 略 }
```

問題2

「10時12分」を「1,0,1,2」と4つの数字で表現する。

「10時2分」を「1,0,-1,2」とする。すなわち、該当する数字がない場合は「0」ではなく「-1」で代用する。

「0時0分」を「-1,0,-1,0」とする。「0,0,0,0」ではない。

このよう4つの数字で時刻を表現するとき、4つの数字が表現する時刻が「0時0分」から「23時59分」の範囲に収まっているかを判定する関数を作成しなさい。たとえば「-1, 0, 1, 0」であれば「0時10分」と解釈できるが、「0, 0, 1, 0」であれば「00時10分」となってしまうので時刻として不正である。同様に「2,4,-1,0」は「24時0分」となり時刻としては正しいが範囲をオーバーするので不正である。

```
static boolean isDateTime(int h1, int h2, int m1, int m2)
```

問題3

数字パーツ（n1, n2, ...）から最大時刻（h時m分）を求める関数を作成しなさい。たとえば [0, 1, 2] という配列が与えられたら [21,0] という配列(21時0分)を返さなければならない。

```
static int[] getMaxDateTime(int[] numbers) { 略 }
```

問題4

1日におけるすべての時刻について最大時刻を調べなさい。そしてそれら最大時刻を出現回数の多い順に表示しなさい。