# Problem Set 2B Hints

**Updated 8 Feb 2022.**

For this question, you don’t actually have to get the midi keyboard Applet up and running.   
Your task is to implement the logic behind the applet. Looking at the text output on the terminal is sufficient.

## Question 1

"When one of these keys is pressed, a note should begin if it isn't already sounding; likewise, when such a key is released, a note should end if it is currently sounding."

In other words, your class should have some way of storing the **Pitch** objects that have been passed to **beginNote**.

Ultimately, think about what happens when you interact with a piano keyboard.

* You press a key down with your finger and then lift it up   
  🡺 you begin a note and end that same note.
* You cannot lift up your finger on a key, if it is not already pressed down   
  🡺 you cannot end a note that has not been begun   
  🡺 if **endNote()** is called on a particular **Pitch()** object, what must you check first?
* You can begin a note if it has not been already been pressed.   
  🡺 If you have pressed down the key with your finger, you cannot press it down one more time, unless you lift it up first.   
  🡺 if **beginNote()** is called, you must check that the same Pitch object has not been previously passed to it.

|  |
| --- |
| midi = Midi.getInstance();  midi.clearHistory();  pm = new PianoMachine();  **//First Test Case**  pm.beginNote(new Pitch(0));  pm.beginNote(new Pitch(0));  Midi.rest(50);  pm.endNote(new Pitch(0));  midi.clearHistory();  **//Second Test Case**  pm.beginNote(new Pitch(0));  Midi.rest(50);  pm.endNote(new Pitch(2));  midi.clearHistory(); |
| Output: |
| on(60,PIANO) rest(50) off(60,PIANO)  on(60, PIANO) |

Note that chords (sounding more than one pitch at the same time) are possible. 

|  |
| --- |
| pm.beginNote(new Pitch(0));  pm.beginNote(new Pitch(2));  Midi.rest(50);  pm.endNote(new Pitch(0));  pm.endNote(new Pitch(2));  midi.clearHistory(); |
| Output: |
| on(60,PIANO) rest(0) on(62,PIANO) rest(50) off(60,PIANO) rest(0) off(62,PIANO) |

## Question 2

Go to **Midi** and notice that **beginNote()** and **endNote()** are overloaded. How would you make use of this fact?

## Question 3

Your attention is brought to the statement "We should be able to shift by two octaves, maximum, in either direction from the starting pitches. " and the relevant code in **TestQ1Hw**.

## Question 4

Consider the purpose of the **NoteEvent** class. 

You will find the following method useful in recording

**System.currentTimeMillis()**

You will find the following method useful in playback

**Thread.sleep()**

When testing your code on your computer, you might find that the resting duration during playback is different from what is specified during recording. I'm not sure what the exact reason for that is, but as long as you use **Thread.sleep()** chances are vocareum will accept your answer.

It would not be necessary to execute **midi.rest()** during playback.

Lastly, remember that our scoring policy is that: full marks is achieved with 430 points out of 460 available.