Pretend you are taking the final exam or a screening for a programmer intern recruitment interview. First, make a fresh directory for this lab. Then, in this directory, start with nothing and perfect a complete Java application (you choose its name) that does the following in the listed order:

1. Make a length 10 array and an array reference variable to refer to it.
   (How and other useful Java fact: `int[] numTimesTyped = new int[10];` It also says that the numbers in a Java array are automatically initialized to 0.)

2. Construct a `java.util.Scanner` from `System.in` so you can use it to get input.
   (Java fact reminder: `java.util.Scanner sc = new java.util.Scanner(System.in);` or save clutter with `import java.util.Scanner;` at the beginning of your code file.)

3. Run a loop where the following is done during each repetition of the body:
   a) Print
      Please type an int 0-9 for the computer to count, or -1 to stop and print the counts:
      or, if your fingers are slow or you are smartly lazy: **0-9 or -1 to stop:**
      (Your notes from Java API documentation say that `System.out.print(something)` will print what `something` means literally or evaluates to and make future printing be on the same line. `println` makes it go to a new line after printing.)
   b) Input one integer which we will name `inNum` and copy it into a variable.
      (Java API notes: `int inNum = sc.nextInt();`)
   c) If `inNum` is negative, make the loop stop and the computer go on to step 5.
      Otherwise, add 1 to the array element in `numTimesTyped at the index given` by `inNum`.
      Your notes show below how to refer to the element in an array named by `numTimesTyped at the location whose index is something`. Of course, each individual element in an array is a variable. You know how to increment the value in a variable.
      `numTimesTyped[ something ]`

Course or your notes say that a good pattern for a loop that repeatedly decides whether to stop or to do something again is:
   `boolean done = false;`  
   `while( ! done )` (which you can also write `while( done == false )`) 
   `{ 
       ....  
       done = true;` (run the operation to change `done`'s value after you decide to stop) 
       (Of course, please `done = true;` inside the `{ .. } body of an if!)
   } 

4. Print how many times each number in the 0-9 range had been typed, following the example:
   **0 was typed 3 times.**
   **1 was typed 7 times.**
   **2 was typed 0 times.**
   etc. up to
   **9 was typed 7 times.**
   (Of course demonstrate that you know how to make a loop do this. Reminder:  
   `System.out.println("Please bla " + N + " ints incl bla bla");`) 

5. Print a cool ASCII histogram, as in the sample screen on the other side of this paper. 

6. (Optional challenge: Print all of your user's favorite numbers.)

For credit, find out (if you don't know already) how to make a (.zip) archive of this lab's directory, and then upload the archive to BB, Lab9Followup. See reverse for our sample.
Optional challenge to print ALL the numbers that were typed the MAXIMUM number of times:

More Java language tips:

```java
if ( TEST ) { ... } and
if ( TEST ) { .... } else { .... }
```

make a decision once.

```java
while( TEST ) { ... } and
for( INIT ; TEST ; UPDATE ) { ... }
```

make decisions repeatedly to make the computer repeat things in a loop.

How to decide which ones to use, and how, in a programming question cannot be summarized in a tip sheet! That's for you to learn by practice and study.