Objectives: Create and use 1D arrays
Deadlines: Challenge 2 Hollywood 8pm tonight (+ MP1 regrade)
Conflict? Email lourent2@`Exam Conflict' **include verifiable reason**

True/False? Q0. A Java array is an object *i.e.* a variable does not actually hold the array; it refers to the array instead. The actual contents of the array are elsewhere in heap memory.
Q1. A Java array can hold a mixture of primitive types. E.g. integer in cell 0, Boolean in cell 1, double value in cell 2.
Q2. The cells (or 'entries') of an array are indexed by an integer.
Q3. The first cell of an array is at index 1. e.g. to add 10 to the first score `scores[1] += 10;`
Q4. `scores.length = 500;` changes the size of the array.
Q5. The last cell in the array 'scores' will be scores[scores.length -1]
Q6. `new int[]{3,5,6,10}` creates an integer array of length 4.
Q7. `new int[99999]` creates a large integer array, each cell is initialized to zero.
Q8. `new char[50]` creates a character array with 50 cells, each cell is initialized to a space.
Q9. `new String[50]` creates a String array with each cell initialized to an empty string
Q10. scores[-1] or scores[scores.length] will produce IndexOutOfBoundsException

Q11 Do I also change `scores[0]` too in the following code? Why?

```
int[] scores = readScores();
String name[]= readNames();
int[] b = scores;
b[0]  = 0;
TextIO.putln( name[0] + " : " + scores[0]);
```

Q12 The bank account swindler:

```
double[] cash = new double[]{-33,102,515,10004,42.07,…};
for(int i = 0; i < cash.length; i ++) {
    if (cash[myAcct] < cash[i] ) {
        cash[myAcct] = cash[i];//Please Fix - It should swap the values.



   } //if
} //for
```

13. What will the following code return?

```
public static String decode() {
   String secret = "Zfxrp";
   char[] mychars = secret.toCharArray();
   String result  = "";
   for (int i = mychars.length-1; i >= 1; i--) {
      char c = mychars[i];
      result += (char) (c - i);
   }
   return result;
}
```

14. What will be the final contents of the array?

```
int [] numbers = new int[] {10, 11, 12, 13};
for (int i = 0; i < numbers.length; i ++)
  numbers[i] = numbers[numbers.length - 1 - i];
```

15. Carefully execute the following code by hand and note the variables values as they change. (i) Determine the final value of each variable.

i:		j:		count:	
result:					

(ii) Execute this code on paper and summarize what it does:

```
int[] arr1 = {10, 20, 30, 40}; //sorted values
int[] arr2 = {18, 20, 25, 99}; //sorted values

int[] result = new int[arr1.length];
int i = 0, j = 0, count = 0;
while (i < arr1.length) {
   if (arr2[j] < arr1[i]) j ++;
   else if (arr2[j] == arr1[i]) i ++;
   else {// must be true that arr2[j] > arr[i]
       result[count] = arr1[i];
       i ++; count ++;
   }
}
```

Extract http://en.wikipedia.org/wiki/Turing_machine : Turing machines are basic abstract symbol-manipulating devices which, despite their simplicity, can be adapted to simulate the logic of any computer algorithm. They were described in 1936 by Alan Turing. Turing machines are not intended as a practical computing technology, but a thought experiment about the limits of mechanical computation. Thus they were not actually constructed. Studying their abstract properties yields many insights into computer science and complexity theory. A Turing machine that is able to simulate any other Turing machine is called a Universal Turing machine (UTM, or simply a universal machine). A Turing Machine consists of:

A **TAPE** which is divided into cells, one next to the other. Each cell contains a symbol from some finite alphabet. A **HEAD** that can read and write symbols on the tape and move the tape left and right one (and only one) cell at a time. In some models the head moves and the tape is stationary. A finite **TABLE** ("action table", or *transition function*) of instructions that, given the *state*(q_i) the machine is currently in *and* the *symbol*(a_j) tells the machine to do the following in sequence (for the 5-tuple models):

* either erase or write a symbol *and then*
* move the head forward or backward *and then*
* assume the same or a *new state* as prescribed

16. PARALLEL ARRAYS: Complete this code to print up to 50 movie titles of movies that grossed over $5m. Print the array index of the highest grossing movie.

```
public static void main(String[] args) {
 double[] gross = …   // gross[i] movie earnings of ith movie (in $m)
 String[] title = …      // title[i] movie title of ith movie.
```

String[] friends= {"A.T.",…,"Roy Campbell",…,"Cinda Heeren"}

17. Replace friend "Roy Campbell" with "Lawrence"
(You'll need to scan through the array)

18. NEW + COPY to append:
You would to add Craig to your friends array. How do you do this?
(Harder: Could you insert a value into the middle?)