A. Helper Methods:

1. Given an integer *array* and indexes *i* and *j*, swap *i*th element with *j*th element. If *I* or *j* is out of bounds, display an error message.

Method name: *swap*

2. Given a double/integer/char array write 3 methods *display* each element tab separated.

Method name: *display*

3. Given *start* and *end* write 2 methods that generate int/double numbers.

Method name: *random*

4. Given a double *value* and int *places*, round the given *value* up to *places* decimal places and returns it.

Method name: *round*

B. Questions:

1. Assign grades.
    a. Write a method *grade* that gets int student *scores* and finds the best score then, assigns grades based on the following scheme returns *grades* char array:

        grade is *A* if score is >= best - 10

        grade is *B* if score is >= best - 20

        grade is *C* if score is >= best - 30

        grade is *D* if score is >= best - 40

        grade is *F* otherwise.

    b. Write a helper method *getMax* that returns the maximum score from given *scores* array

2. Eliminate duplicates
    a. Write a method *eliminateDuplicates* that returns a new array by eliminating duplicates in an int array. Order of appearances for numbers must not change

3. Shuffle the array
    a. Write a method *shuffle* that takes an int array and shuffles it randomly

4. Locker puzzle
    a. A school has 100 lockers and 100 students. All lockers are closed on the first day of school. As the students enter, the first student, denoted as S1, opens every locker. Then the second student, S2, begins with the second locker, denoted L2, and closes every other locker (every second locker). Student S3 begins with the third locker, L3, and changes every third locker (closes it if it was open, and opens it if it was closed). Student S4 begins with L4 and changes every fourth locker. S5 starts with L5 and changes every fifth locker, and so on, until student S100 changes L100.

       After all the students have passed through the building and changed the lockers, which lockers are open? Find the indices of lockers that are open and return them as counting numbers (this means index 0 should be 1, so add +1 to each index value you are storing in returned array).

       Write this method *lockers* that takes Boolean array that will have length of 100 and returns indices of open lockers as counting numbers.