

Solution to Practice Problems: Array Problems

Repeat-X and Sum Algorithms with Arrays

Write a short Java program to solve each of the following problems. Each one will involve an array, plus a Repeat-X or an accumulate algorithm (and maybe more than one) --- it's up to you to figure out how!

1. Read 10 ints from the keyboard, and store them in an array. For each int in the array, display a line with that many "*" characters.

```
public class PrintLines {
    public static void main(String [] args) {
        Scanner kb = new Scanner(System.in);

        // Repeat-X algorithm to read in 10 ints and store in an array
        int [] numStars = new int[10];
        for(int i=0; i<numStars.length; i++) {
            numStars[i] = kb.nextInt();
        }

        // Repeat-X algorithm to print 10 lines
        for(int i=0; i<numStars.length; i++) {

            // Repeat-X algorithm to print numStars[i] "*"s
            for(int j=0; j<numStars[i]; j++) {
                System.out.print("*");
            }

            System.out.println();
        }
    }
}
```

2. Read in 10 ints from the keyboard, and store them in an array. Find the maximum and minimum values in the array, and display them on the screen.

```
public class PrintMaxMin {
    public static void main(String [] args) {
        Scanner kb = new Scanner(System.in);

        // Repeat-X algorithm to read in 10 ints and store in an array
        int [] intArr = new int[10];
        for(int i=0; i<intArr.length; i++) {
            intArr[i] = kb.nextInt();
        }

        // Accumulate algorithm to find maximum element
        int max = intArr[0];
        for(int i=1; i<intArr.length; i++) {
            if(intArr[i] > max) {
                max = intArr[i];
            }
        }
    }
}
```

```

    }

    System.out.println("max = " + max);

    // Accumulate algorithm to find minimum element
    int min = intArr[0];
    for(int i=1; i<intArr.length; i++) {
        if(intArr[i] < min) {
            min = intArr[i];
        }
    }

    System.out.println("min = " + min);
}
}

```

3. Read in 10 ints from the keyboard, and store them in an array. Find the *position* (or *index*) of the maximum and minimum values in the array, and *swap* them (move the biggest element to the position of the smallest, and move the smallest element to the position of the biggest).

```

// this is just to allow the Arrays.toString() command at the end
// it's not necessary to solve the problem; just a convenience
import java.util.Arrays;

public class PrintMaxMin {
    public static void main(String [] args) {
        Scanner kb = new Scanner(System.in);

        // Repeat-X algorithm to read in 10 ints and store in an array
        int [] intArr = new int[10];
        for(int i=0; i<intArr.length; i++) {
            intArr[i] = kb.nextInt();
        }

        // Accumulate algorithm to find position of max element
        // this is just like the earlier problem, but now we need to
        // keep track of max AND the position of max in the array
        int max = intArr[0];
        int maxPos = 0; // maxPos starts at position 0
        for(int i=1; i<intArr.length; i++) {
            if(intArr[i] > max) {
                max = intArr[i];
                maxPos = i;
            }
        }

        // Accumulate algorithm to find position of min element.
        // This is just like the earlier problem, but now we need to
        // keep track of min AND the position of min in the array
        int min = intArr[0];
        int minPos = 0;
        for(int i=1; i<intArr.length; i++) {
            if(intArr[i] < min) {
                min = intArr[i];
            }
        }
    }
}

```

```

        minPos = i;
    }
}

// swap the biggest and smallest elements
intArr[minPos] = max;
intArr[maxPos] = min;

// here's a handy command to print out the entire array
System.out.println(Arrays.toString(intArr));
}
}

```

4. Read 10 ints from the keyboard, and store them in an array. Display "true" on the screen if there is an even number of even numbers among these 10. Otherwise, display "false".

```

public class EvenEvens {
    public static void main(String [] args) {
        Scanner kb = new Scanner(System.in);

        // Repeat-X algorithm to read in 10 ints and store in an array
        int [] intArr = new int[10];
        for(int i=0; i<intArr.length; i++) {
            intArr[i] = kb.nextInt();
        }

        // Accumulate algorithm to find number of even elements
        int countEven = 0;
        for(int i=0; i<intArr.length; i++) {
            if(intArr[i] % 2 == 0) {
                countEven++;
            }
        }

        if(countEven % 2 == 0) { // even number of evens
            System.out.println("true");
        } else {
            System.out.println("false");
        }
    }
}

```