MIDTERM CSE143x SOLUTIONS WINTER 2015

1) skip likes to drive with george
connor likes to skip with shan
chris likes to work with drive
john likes to dance with shan

2)
4 4
3 2 4 5
3 3
3 2 4 5

3)
N S S
S A S
N S S
A S S
S N S

4.)

```
public static int digitRange(int n) {
    int min = n % 10;
    int max = n % 10;
    while (n > 0) {
        int digit = n % 10;
        n = n / 10;
        if (digit < min) {
            min = digit;
        }
        if (digit > max) {
            max = digit;
        }
    }
    return max - min;
}
```

5)

```
public static boolena hasAlternatingParity(int[] arr) {
	boolean alt = true;
	for (int i = 0; i < arr.length - 1; i++) {
		if (arr[i] % 2 == arr[i+1] % 2) {
			alt = false;
		}
	}
	if (!alt) {
		for (int i = 0; i < arr.legnth; i++) {
			arr[i] = 0;
		}
	}
	return alt;
```

6)

```
public static void printDuplicates(Scanner input) {
	while (input.hasNextLine()) {
		String line = input.nextLine();
		Scanner lineScan = new Scanner(line);
		String token = lineScan.next();
		 int count = 1;
		 while (lineScan.hasNext()) {
			String token2 = lineScan.next();
			if (token2.equals(token)) {
				count++;
			} else {
				if (count > 1) {
					System.out.print(token + "*" + count + " ");
				}
				 token = token2;
				 count = 1;
			}
		 }
		 if (count > 1) {
			System.out.print(token + "*" + count);
		}
		 System.out.println();
	}
}
```

```
public static void printDuplicates(Scanner input) {
	while (input.hasNextLine()) {
		String line = input.nextLine();
		Scanner lineScan = new Scanner(line);
		String token = lineScan.next();
		int count = 1;
		while (lineScan.hasNext()) {
			String token2 = lineScan.next();
			if (token2.equals(token)) {
				count++;
			}

			if (count > 1 && (!lineScan.hasNext() || !token2.equals(token)))
				System.out.print(token + "*" + count + " ");
				count = 1;
			token = token2;
		}
		 System.out.println();
	}
}
```

7)

```
public static int numMissing(int[] vals) {
	int total = 0;
	int sum = 0;
	for (int i = 1; i <= vals.length + 1; i++) {
		total += i;
	}
	for (int i= 0; i <vals.length; i++) {
		sum+= vals[i];
	}
	return total – sum;
}
```