

## Fundamentals of CS Final – Revision

### Arrays

For this section, and any other Array sections in this packet, feel free to look up how to use ArrayLists in Java. They may be easier than the Arrays we've covered in class.

Google is your friend, but you can start [here](#).

1. Write a function that takes in an int n and returns an Array/ArrayList.  
The function should create int array called nums with 40 elements.  
The array should consist of the numbers from 0 – n where each element is 10 more than the last (i.e.  $n - (n-1) = 10$ ).

2. Given the following array:

```
int[] a = {-1, -2, -6, 3, 7, 8, 23, 4, 23, 12, -10}
```

Write a loop to do each of the following:

- a. Add 1 to every element of a.
- b. Count the number of negative numbers in a.
- c. Create a new array b which is the same size as a, and copy all elements from a into b.
- d. Print the elements of a in reverse order.
- e. Count the number of elements in a that have values between 10 and 20 inclusive.

3. Given the declarations

```
int [] sample=new int[8];  
int i, k;
```

Show the contents of the array sample after the following code segment is executed.

Use a question mark to indicate any undefined values in the array.

```
for (k=0; k<8;k++) {  
    sample[k] = 10 - k;  
}
```

### Methods

1. Write a method multiple that takes in 2 ints and returns a boolean. It determines, for a pair of integers, whether the second integer is a multiple of the first. The method should return true if the second is a multiple of the first, and return false otherwise.

2. Write a method called `alarm` that prints the word "Alarm!" multiple times on separate lines. The method should accept an integer parameter that specifies how many times the output line is printed. This method returns nothing.
3. Write a method called `sum100` that returns the sum of the integers from 1 to 100.
4. Write a method called `multiConcat` that takes a `String` and an `int` as parameters, and returns a `String` that is the parameter string concatenated with itself `n` number of times (where `n` is the second parameter).

For example, if the parameters are "hi" and 4, the return value is "hihihihi".

5. Write a method called `reverse` that accepts a `String` as a parameter and returns a `String` that contains the characters of the parameter in reverse order. Note: there is actually a method in the `String` class that performs this operation, but for the sake of this exercise you will write your own.

## Recursion

P.S. For #3 and #4, DON'T USE LOOPS. You have to write recursive methods.

To show that you understand how to trace a recursive method, you **must** show the recursive calls that each invocation would make. For example, suppose the mystery method is as follows:

```
public static int mystery0(int x, int y) {  
    if (x == 0 || y == 0)  
        return 0;  
    else  
        return x + mystery0(y-1, x);  
}
```

Given the call to `mystery(3, 2)` your answer would be

```
mystery0(3, 2) = 3 + mystery0(1, 3)  
               = 3 + 1 + mystery0(2, 1)  
               = 3 + 1 + 2 + mystery0(0, 2)  
               = 3 + 1 + 2 + 0 = 6
```

It multiplies

1. Consider the following recursive mystery method below.

```
public static int mystery1(int a, int b) {  
    if (a < b) {  
        return 0;  
    } else {  
        return 1 + mystery1(a - b, b);  
    }  
}
```

- a. What is the value returned for each of the following method calls? Show the recursive method calls that are generated.
- i. `mystery1(7, 2);`
  - ii. `mystery1(29, 5);`
  - iii. `mystery1(5, 7);`
- b. In just a few words, state what the mystery method is computing assuming  $a > 0$  and  $b > 0$  for the initial call to mystery.

2. Consider the following recursive mystery method below.

```
public int mystery2(int x, int y) {  
    if (x < 0) {  
        return -mystery2(-x, y);  
    } else if (y < 0) {  
        return -mystery2(x, -y);  
    } else if (x == 0 && y == 0) {  
        return 0;  
    } else {  
        return 100 * mystery2(x / 10, y / 10) + 10 * (x %  
10) + y % 10;  
    }  
}
```

- a. What is the value returned for each of the following method calls? Show the recursive method calls that are generated.
- i. `mystery2(7, -2);`
  - ii. `mystery2(29, 45);`
  - iii. `mystery2(135, 246);`

b. In just a few words, state what mystery method is returning.

3. Write a method `isBackwards` that has two `String` parameters and returns `true` if the two strings have the same sequence of characters (ignoring whitespace) but in the opposite order and returns `false` otherwise.

P.S. You might find the `isWhitespace` method of the `Character` class useful.

For example,

```
isBackwards("fried", "deirf") -> true
```

```
isBackwards("sit", "toes") -> false
```

```
isBackwards("", "") -> true
```

```
isBackwards("I madam", "mad am I") -> true
```

4. Write a recursive method `sumDigits` that has one `int` parameter and returns the sum of the digits in the integer specified. Remember, your method should not use loops.

For example, if the integer is 15121, then this method should return 10.

## Classes

I'm lazy, so for this one, I'm going to ask you to finish the `Library` class that we had started in class.

See skeleton [here](#).

Feel free to look up how to use `ArrayLists`; this might make your job easier.