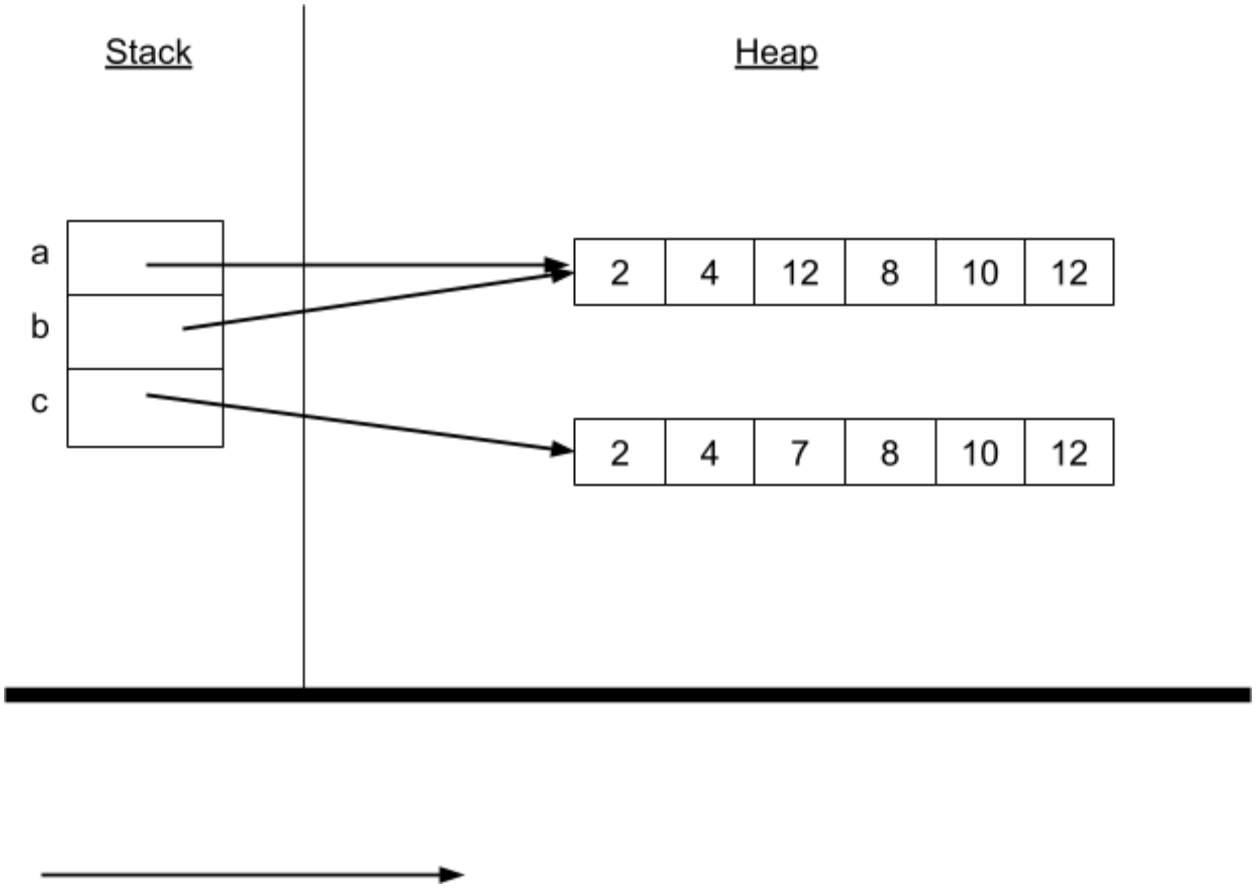
Problem Set 1, Part I

Problem 1: Memory management and arrays

1-1)



1-2)

12 12 7

Problem 2: Array practice

2-1)

```
public static boolean isSorted(int[] arr) {

      if (arr == null){
            throw new IllegalArgumentException();
      }
      if (arr.length == 0){
```

```
            return true;
      }

      for (int i = 1; i < arr.length; i++){
            if (arr[i] < arr[i-1]){
                  return false;
            }
      }

      return true;

}
```

2-2)

```
public static void scale(int[] arr, int factor) {

      if (arr == null){
            throw new IllegalArgumentException();
      }
      if (arr.length == 0){
            return;
      }

      for (int i = 0; i < arr.length; i++){
            arr[i] = arr[i]*3;
      }

}
```

Problem 3: Recursion and the runtime stack

```
3-1)
mystery(20, 6)
-------------
      a = 20
      b = 6
      myst_rest = mystery(14, 6) = 7
      return 2 + 7 = 9

      mystery(14, 6)
      -----------------
      a = 14
      b = 6
      myst_rest = mystery(8, 6) = 5
      return 2 + 5

      mystery(8, 6)
      -----------------
      a = 8
      b = 6
      myst_rest = mystery(2, 6) = 3
      return 2+3

      mystery(2, 6)
      -----------------
      a = 2
      b = 6
      myst_rest = mystery(-4, 6) = 1
      return 2+1

      mystery(-4, 6)
      -----------------
      a = -4
      b = 6
      return 1

3-2) 9

3-3) 6  (5 mystery() and 1 main())
```

3-4)when b <= 0, the infinite loop occurs because a - b doesn’t change or always gets a positive number. Thus, there is no base case that a < 0.

Problem 4: Using recursion

```
public static void printReverse(Object[] arr, int i) {
     if (arr == null || i < 0) {
          return;
     }
     System.out.println(arr[i]);
     printReverse(arr, i-1);

}
```