

Harry Langford hjel2

```
4. a) class MalfunctioningDeviceException extends Exception {
    String
    final String device;
    final String malfunction;
    public MalfunctioningDeviceException(String device,
                                         String malfunction) {
        this.device = device;
        this.malfunction = malfunction;
    }
}
```

```
b) class Device {
    String name
    Map<String, Integer> settings;
    public Device(String dname) {
        name = dname;
        settings = new HashMap<String, Integer>();
    }
    public void store(String setting, int value) throws
        MalfunctioningDeviceException {
        if (value < 0 || value > 100) {
            throw new DeviceMalfunctioningDeviceException(
                name, "Invalid setting passed");
        }
        settings.put(setting, value);
    }
    public void int get(String setting) throws MalfunctioningDeviceException
        -Exception {
        if (settings.get(setting) is null) { return -1;
            throw new MalfunctioningDeviceException(name,
                "Device has no such setting");
        }
    }
}
```

Harry Langford hjel2

```
}  
return Settings.get(setting);  
}
```

b) i)

```
class Printorder {  
    public static Device[] print (String setting,  
                                   Device[] devices) {  
        return List.of(devices).stream().compare(  
            Device::get(setting)).reverse().toArray();  
    }  
}
```

ii)

Harry Langford hje12

```
c) class Cypher {  
    public static decode (String Cyphertext) {  
        int[] count = new int[26];  
        for (s : cyphertext) {  
            count[s-41]  
            count[(int) s - 41] ++;  
        }  
        int maxocc = 0;  
        int pos = 0;  
        for (int i = 0; i < 27; i++) {  
            if (count[i] > maxocc) {  
                maxocc = count[i];  
                pos = i;  
            }  
        }  
        return (pos - 4);  
    }  
}
```

d) ~~The algorithm to do this is recursive. It starts for each ^{index} grid. sees if the rectangle sum~~

```
class Sums {  
    public static sumtozero (int [][arr]) {  
        int total = 0;  
        for (int xend  
            for (int xend = 0; xend < arr.length; xend++) {  
                for (int xstart = 0; xstart < xend; xstart++) {  
                    for (int yend = 0; yend < arr[0].length; yend++) {  
                        for (int ystart = 0; ystart < yend; ystart++) {  
                            for (int x = xstart; x < xend; x++) {  
                                for  
                                    int sum = 0;  
                                    for (int x = xstart; x < xend; x++) {
```


HongLangford hje12

```
for(int y = ystart; y <= yend; y++) {  
    sum += arr[x][y];  
}  
}  
if(sum == 0) {  
    total += 1;  
}  
}  
}  
}  
}  
return total;  
}  
}
```