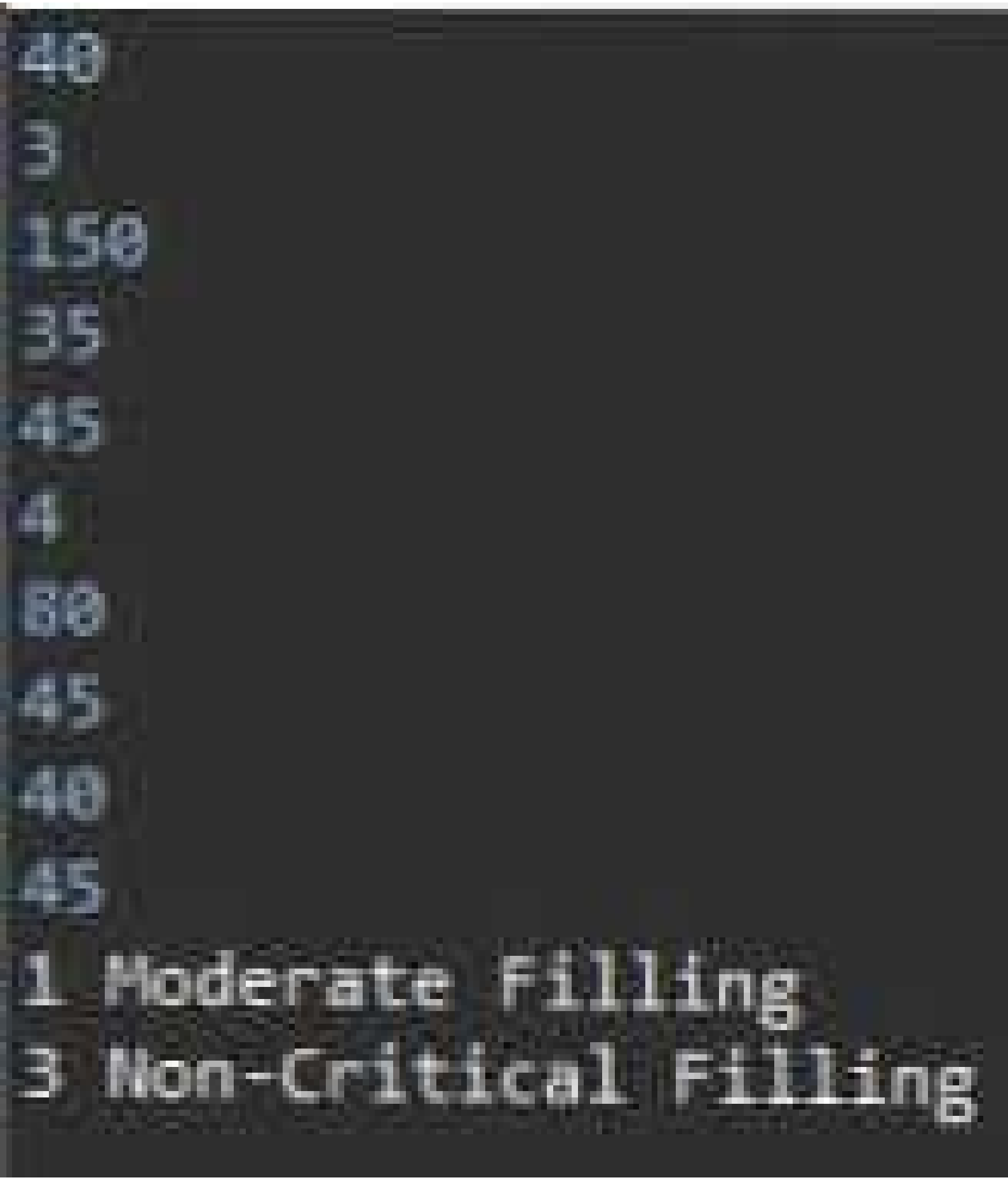
```java
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

@SuppressWarnings("unused")
class Inventory {
    private String inventoryId;
    private int maximumQuantity;
    private int currentQuantity;
    private int threshold;

    public Inventory(String inventoryId, int maximumQuantity, int currentQuantity, int threshold) {
        this.inventoryId = inventoryId;
        this.maximumQuantity = maximumQuantity;
        this.currentQuantity = currentQuantity;
        this.threshold = threshold;
    }

    // Getters and Setters
    public String getInventoryId() {
        return inventoryId;
    }

    public void setInventoryId(String inventoryId) {
        this.inventoryId = inventoryId;
    }

    public int getMaximumQuantity() {
        return maximumQuantity;
    }

    public void setMaximumQuantity(int maximumQuantity) {
        this.maximumQuantity = maximumQuantity;
    }

    public int getCurrentQuantity() {
        return currentQuantity;
    }

    public void setCurrentQuantity(int currentQuantity) {
        this.currentQuantity = currentQuantity;
    }

    public int getThreshold() {
        return threshold;
    }

    public void setThreshold(int threshold) {
        this.threshold = threshold;
    }
}
```

```java
package javaassessment;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
class Main {
    public static Inventory[] replenish(Inventory[] inventories, int limit) {
        List<Inventory> replenishedInventories = new ArrayList<>();
        for (Inventory inventory : inventories) {
            if (inventory.getThreshold() >= limit) {
                replenishedInventories.add(inventory);
            }
        }
        return replenishedInventories.toArray(new Inventory[0]);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Inventory[] inventories = new Inventory[4];

        for (int i = 0; i < 4; i++) {
            String inventoryId = scanner.next();
            int maximumQuantity = scanner.nextInt();
            int currentQuantity = scanner.nextInt();
            int threshold = scanner.nextInt();
            inventories[i] = new Inventory(inventoryId, maximumQuantity, currentQuantity, threshold);
        }

        int limit = scanner.nextInt();

        Inventory[] replenishedInventories = replenish(inventories, limit);

        for (Inventory inventory : replenishedInventories) {
            if (inventory.getThreshold() > 75) {
                System.out.println(inventory.getInventoryId() + " Critical Filling");
            } else if (inventory.getThreshold() >= 50 && inventory.getThreshold() <= 75) {
                System.out.println(inventory.getInventoryId() + " Moderate Filling");
            } else {
                System.out.println(inventory.getInventoryId() + " Non-Critical Filling");
            }
        }

        scanner.close();
    }
}
```


40
3
150
35
45
4
80
45
40
45
1 Moderate Filling
3 Non-Critical Filling