Task1

1、整型：byte short int long 符号型:char 浮点型：float double 布尔型：boolean

2、byte:1字节 -127 - 128 short:2字节 -32768 - 32767 int:4字节 -2的31到2的31次方-1

long：8字节 -2的63次方到2的63次方-1

3、隐性类型转换 b的值是52 c是字符类型被赋值’0’字符在参与数学运算时，会自动转换成其所对应的数字编码，而’0’对应的编码是48

所以b=a+c=4+48=52

4、//false， new每次创建的都是一个新的实列，不会复用缓存，所以x和y的地址不同，所以不等

//true valueof使用的是Integer缓存池，z和k指向缓存池内同一个对象，内存地址相同，==比较情况相同

//false Integer缓存池只能储存-127到128的值，300超出了范围，则会在堆中创建新的对象，导致m和p的地址不同

Task2

5、c=13,++a是先将a加1过后再进行运算b++是再运算后再给b+1.所以最后a是6，b是8

6、int的最高位是符号位0表示正数1表示负数，剩下31位是数值位表示数值大小

float的最高位也是符号位0表示正数1表示负数，有8位指数用于表示指数的大小，后23位是尾数位表示有效数字的小数部分

当两个正数相加超过类型的范围时,会发生溢出，导致首位数字变为1，而两个正数的和就变为了负数。

Task3
第一，二题如下

```
1. import java.util.*;
2.
3. public class Collection01 {
4.     // 使用HashMap存储科目和错题数量的映射
5.     private Map<String, Integer> wrong = new HashMap<>();
6.
7.     public static void main(String[] args) {
8.         Collection01 system = new Collection01();
9.         String rawData =
   "math:5,English:10,Chinese:10,math:20,English:10,chemistry:30,math:10,math:20";
10.         system.processRawData(rawData);
11.         system.display();
```

```
    }

    public void processRawData(String rawData) {
        String[] entries = rawData.split(",");
        for (String entry : entries) {
            String[] parts = entry.split(":");
            if (parts.length == 2) {
                String subject = parts[0].trim();
                try {
                    int errorCount = Integer.parseInt(parts[1].trim());
                    // 使用HashMap的put和getOrDefault方法统计错题
                    wrong.put(subject, wrong.getOrDefault(subject, 0) +
errorCount);
                } catch (NumberFormatException e) {
                    System.out.println("数据格式错误: " + entry);
                }
            }
        }
    }

    private void display() {
        System.out.println("=== 错题统计结果（按错题数排序）===");
        wrong.entrySet().stream()
                .sorted(Map.Entry.<String, Integer>comparingByValue().reversed())
                .forEach(entry ->
                        System.out.println(entry.getKey() + ": " +
entry.getValue() + "题"));
    }
}
`
```

第三题:

```
import java.util.*;

public class Collection01 {
// 使用HashMap存储科目和错题数量的映射
private Map<String, Integer> wrong = new HashMap<>();

    public static void main(String[] args) {
        Collection01 system = new Collection01();
        String rawData =
"math:5,English:10,Chinese:10,math:20,English:10,chemistry:30,math:10,math:20";
        system.processRawData(rawData);
        system.display();
        system.Add();
```

```
13.
14.     }
15.
16.     public void processRawData(String rawData) {
17.         String[] entries = rawData.split(",");
18.         for (String entry : entries) {
19.             String[] parts = entry.split(":");
20.             if (parts.length == 2) {
21.                 String subject = parts[0].trim();
22.                 try {
23.                     int errorCount = Integer.parseInt(parts[1].trim());
24.                     // 使用HashMap的put和getOrDefault方法统计错题
25.                     wrong.put(subject, wrong.getOrDefault(subject, 0) +
    errorCount);
26.                 } catch (NumberFormatException e) {
27.                     System.out.println("数据格式错误: " + entry);
28.                 }
29.             }
30.         }
31.     }
32.
33.     private void display() {
34.         System.out.println("=== 错题统计结果（按错题数排序）===");
35.         wrong.entrySet().stream()
36.                 .sorted(Map.Entry.<String, Integer>comparingByValue().reversed())
37.                 .forEach(entry ->
38.                         System.out.println(entry.getKey() + ": " +
    entry.getValue() + "题"));
39.     }
40.     public void Add() {
41.         Scanner scanner = new Scanner(System.in);
42.         System.out.println("错题统计系统启动，请输入数据（格式：科目:错题数），输
    入'quit'退出");
43.
44.         while (true) {
45.             System.out.print("请输入数据: ");
46.             String input = scanner.nextLine().trim();
47.
48.             if ("quit".equalsIgnoreCase(input)) {
49.                 break;
50.             }
51.
52.             if (processSingleInput(input)) {
53.                 System.out.println("数据添加成功！");
54.             } else {
55.                 System.out.println("输入格式错误，请重新输入！");
```

```
            }
        }

        display();
    }

    private boolean processSingleInput(String input) {
        try {
            String[] parts = input.split(":");
            if (parts.length != 2) return false;

            String subject = parts[0].trim();
            int errorCount = Integer.parseInt(parts[1].trim());

            wrong.put(subject, wrong.getOrDefault(subject, 0) + errorCount);
            return true;
        } catch (NumberFormatException e) {
            return false;
        }
    }
}
`
```

第四题:

```
import java.util.*;

public class Collection01 {
private Map<String, Map<String, Integer>> studentsWrong = new HashMap<>();

    public static void main(String[] args) {
        Collection01 system = new Collection01();

        String xiaomingData =
"math:5,English:10,Chinese:10,math:20,English:10,chemistry:30,math:10,math:20";
        system.processStudentData("小明", xiaomingData);

        String xiaohongData = "math:8,Chinese:15,English:12,chemistry:10";
        system.processStudentData("小红", xiaohongData);

        system.displayStudent("小明");
        system.displayStudent("小红");
    }

    public void processStudentData(String studentName, String rawData) {
        Map<String, Integer> subjects = studentsWrong.computeIfAbsent(
```

```
                studentName, k -> new HashMap<>());

        String[] entries = rawData.split(",");
        for (String entry : entries) {
            String[] parts = entry.split(":");
            if (parts.length == 2) {
                String subject = parts[0].trim();
                try {
                    int errorCount = Integer.parseInt(parts[1].trim());
                    subjects.put(subject, subjects.getOrDefault(subject, 0) +
errorCount);
                } catch (NumberFormatException e) {
                    System.out.println("数据格式错误: " + entry);
                }
            }
        }
    }

    private void displayStudent(String studentName) {
        Map<String, Integer> subjects = studentsWrong.get(studentName);
        if (subjects == null) {
            System.out.println(studentName + " 暂无错题数据");
            return;
        }

        System.out.println("=== " + studentName + " 的错题统计结果（按错题数排序）
===");
        subjects.entrySet().stream()
                .sorted(Map.Entry.<String, Integer>comparingByValue().reversed())
                .forEach(entry ->
                        System.out.println(entry.getKey() + ": " +
entry.getValue() + "题"));
        System.out.println();
    }

    public void processRawData(String rawData) {
        processStudentData("小明", rawData); // 默认处理小明的数据
    }

    private void display() {
        displayStudent("小明"); // 默认显示小明的统计
    }
}
```