Java03

Task1

Q1:

整型：byte,short,int,long

字符型：char

浮点型：float，double

布尔型：boolen

Q2:

byte:　1字节　　-128~127

short：2字节 -32768~32767

int:　　4字节 约-21亿~21亿

long:　8字节　　约-922亿~922亿

Q3:

输出的值是52

是隐式类型转换

因为 '0' 转换成整数后是它的ascII码对应的数值，经查阅， '0' 对应的是48，故48 + 5 = 52

Q4：

Integer是int的包装类

第一部分报错，但结果是false

第二部分为true

第三部分为false

因为第一部分new了两个对象，地址不同，输出为false。

第二部分中由于valueof的存在，k和z都指向同一个缓存的数值，故输出为true

第三部分的数据300超出了缓存，所以又new了两个对象，和第一个一样

Task2

Q5:

输出结果是：

13

6 8

++a是说a先加一再代入运算，b++是说b先代入运算后自加

Q6：

查阅资料后得知，int类型第一位用于表示符号，其后用于记录大小

float类型使用指数近似表示

当数据越界之后，会发生数据溢出，而返回负数

Task3

Q1:

基础版

```
public class Task3 {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("Math", 0);
        map.put("English", 0);
        map.put("Chinese", 0);
        map.put("Chemistry", 0);
        int m = map.get("Math");
        int e = map.get("English");
        int c = map.get("Chinese");
        int ch = map.get("Chemistry");
        m += 5;
        e += 10;
        c += 10;
        m += 20;
        e += 10;
        ch += 30;
        m += 10;
        m += 20;
        map.put("Math", m);
        map.put("English", e);
        map.put("Chinese", c);
        map.put("Chemistry", ch);
        List<Map.Entry<String, Integer>> list = new ArrayList<>(map.entrySet());
        Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
            @Override
```

```
            public int compare(Map.Entry<String, Integer> o1, Map.Entry<String,
Integer> o2) {
                return o2.getValue().compareTo(o1.getValue()); // 降序排列
            }
        });

        // 输出排序结果
        for (Map.Entry<String, Integer> entry : list) {
            System.out.println(entry.getKey() + ": " + entry.getValue());
        }
    }
}
```

Q3:

可以自己加入数据的加强版本!

```
package task3;

import java.util.*;

public class Task3 {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("Math", 0);
        map.put("English", 0);
        map.put("Chinese", 0);
        map.put("Chemistry", 0);
        int m = map.get("Math");
        int e = map.get("English");
        int c = map.get("Chinese");
        int ch = map.get("Chemistry");
        Scanner sc = new Scanner(System.in);
        String choice = "yes";
        while (choice.equals("yes")){
            System.out.println("请输入学科：(首字母大写)");
            String subject = sc.next();
            System.out.println("请输入数量：");
            int number = sc.nextInt();
            switch (subject) {
                case "Math":
                    m += number;
                    break;
                case "English":
                    e += number;
                    break;
                case "Chinese":
                    c += number;
                    break;
                case "Chemistry":
                    ch += number;
                    break;
                default:
                    System.out.println("输入错误！");
            }
```

```
            System.out.println("请输入是否继续：(yes/no)");
            choice = sc.next();
        }

        map.put("Math", m);
        map.put("English", e);
        map.put("Chinese", c);
        map.put("Chemistry", ch);
        List<Map.Entry<String, Integer>> list = new ArrayList<>(map.entrySet());
        Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
            @Override
            public int compare(Map.Entry<String, Integer> o1, Map.Entry<String,
Integer> o2) {
                return o2.getValue().compareTo(o1.getValue()); // 降序排列
            }
        });

        // 输出排序结果
        for (Map.Entry<String, Integer> entry : list) {
            System.out.println(entry.getKey() + ": " + entry.getValue());
        }
    }
}
```

关于小红的实现，我打算写一个学生类，并把整个方法变成一个成员方法，new两个对象，分别调用这个方法即可~~~但是真的不想写了，后面还有很多题呢QWQ