

学号	姓名	提交时间	总分	成绩
20225102050109	许钰野	2024-03-28 09:46:22	100	40

1. 【填空题】下面是冒泡排序的代码片段，对数组元素进行降序排列。 得分：20/30

```
int[] a = {5,4,2,3,1};
//i用于控制比较的轮数
for(int i=0;i<_____;i++){
    //j用来表示数组的下标，每一轮j都是从0开始，并且和紧邻的数组开始比较
    for(int j=0;j<_____;j++){
        //使用两个紧邻的元素比较，对应的下标和j+1
        if(_____) {
            //交换
            int temp = a[j]; // 现将j位置对应的元素，存到一个临时变量里
            a[j] = a[j+1]; // 将j+1位置上的元素赋值到j位置上
            a[j+1] = temp; // 将临时变量里的元素放置到j+1这个位置
        }
    }
}
System.out.println(Arrays.toString(a)); // [ 5,4,3,2,1]
```

学生答案：1.a.length-1
2.a.length-1-i
3.a[j]>a[j+1]

参考答案：1.a.length-1
2.a.length-1-i
3.a[j]<a[j+1]

2. 【填空题】定义方法的格式如下： 得分：10/10

```
修饰词 _____ 方法名(形式参数){
    方法体
}
```

学生答案：1.返回值类型

参考答案：1.返回值类型

3. 【填空题】//int[][] as = {{1,2,3},{4,5},{6,7,8,9}}; 得分：0/10

```
int[][] as = new int[3][];
as[0] = new int[]{1,2,3};
as[1] = new int[]{4,5};
as[2] = new int[]{6,7,8,9};
System.out.println(_____); // 5
```

学生答案：1.a[1][1]

参考答案：1.as[1][1]

4. 【简答题】使用Math.random()随机一个整数n，范围[4,6]，用作两个数组names，scores的长度。 得分：10/50

names是String类型的数组，用于存储从控制台扫描进来的n个学生的名称
scores是int类型的数组，用于存储这些学生的数学成绩。
计算出平均分avg，最高分max，最低分min。

学生答案：

```
public static void main(String[] args) {
    Random r = new Random();
    int n = r.nextInt( bound: 6)+5;

    String[] names = new String[n];
    Scanner scan = new Scanner(System.in);
    for (int i =0 ;i<n;i++){
        System.out.println("请输入第"+(i+1)+"个学生的姓名: ");
        names [i]= scan.next();
    }

    System.out.println("学生姓名: "+Arrays.toString(names));
}
```

```
参考答案 : public static void main(String[] args) {
    int n = (int)(Math.random()*3)+4;
    String[] names = new String[n];
    int[] scores = new int[n];
    Scanner scan = new Scanner(System.in);
    for (int i = 0;i<names.length;i++){
        System.out.println("请输入第"+(i+1)+"个学生的名称");
        names[i] = scan.next();
        System.out.println("请输入第"+(i+1)+"个学生的成绩");
        scores[i] = scan.nextInt();
    }
    System.out.println(Arrays.toString(names));
    System.out.println(Arrays.toString(scores));
    int sum = -1;
    int max = -1;
    int min = 1000;
    for(int i = 0;i<scores.length;i++){
        sum+=scores[i];
        if(max<scores[i]){
            max = scores[i];
        }
        if(min>scores[i]){
            min = scores[i];
        }
    }
    System.out.println("平均分:"+sum*1.0/scores.length);
    System.out.println("最高分:"+max);
    System.out.println("最低分:"+min);
}
```