

1, 以下关于 sql 查询语句执行顺序描述正确的是: ( )

A select->from->where->group by->having->order by

B from->where->select->group by->having->order by

C from->where->group by->having->select->order by

D select->from->group by->having->where->order by

2, 一位老师有 2 个推理能力很强的学生, 他告诉学生他手里有以下的牌:

黑桃: 2, 5, 7, 9, J, K

红心: 3, 4, 9, J, K

梅花: 5, 8, 9, Q

方块: 2, 7, 8

然后从中拿出一张牌, 告诉 A 这张牌的大小, 告诉了 B 这张牌的花色;

A: 我不知道这张是什么牌

B: 我就知道你肯定不知道这张是什么牌

A: 现在我知道

B: 现在我也知道了

请问这张是什么牌? ( )

A 方片 8

B 梅花 5

C 红心 k

D 黑桃 7

3, 下列排序算法在最好情况下的时间复杂度为  $O(n)$  的是()(多选)

A 堆排序 B 快速排序 C 冒泡排序 D 插入排序 E 桶排序

4, (多选) 如果一个二叉树的先序遍历是: DCAGFEB, 那么它的中序遍历可能是: ()

A.ACGDEFB

B.CDFGEAB

C.DCGAFEB

D.CGAFDEB

5, 某小红薯在小红书的活动中抽奖中了一定价值的薯券, 这些薯券可以用来购买一批商品, 求有多少种购买组合。其中一件商品可以买多件。

输入: 薯券金额、商品分别价格

输出: 组合数

答: `import java.util.*;`

`public class Main {`

`public static void main(String[] args) {`

`Scanner in = new Scanner(System.in);`

`int n = in.nextInt();`

`String s = in.next();`

```

String[] tmp = s.substring(1, s.length() - 1).split(",");
int[] arr = new int[n + 1];
int[] p = new int[tmp.length];
for(int i = 0; i < tmp.length; i++) {
    p[i] = Integer.parseInt(tmp[i]);
}

arr[0] = 1;

for(int a : p) {
    for(int i = a; i < arr.length; i++) {
        arr[i] = (arr[i] + arr[i - a]);
    }
}

System.out.println(arr[n]);
}
}

```

6, 薯队长带着小红薯参加密室逃脱团建游戏, 首先遇到了反转游戏, 小红薯们根据游戏提示收集了多个单词线索, 并将单词按要求加一个空格组 成了句子, 最终要求把句子按单词反转解密。 说明: 收集的时候单词前后可能会有多个空格, 反转后单词不能有多余空格, 具体见输入输出样例。

答:

```

import java.util.*;
public class Main{
public static void main(String[] args)
{
    Scanner in=new Scanner(System.in);
    String str=in.nextLine();
    List<String> words=Arrays.asList(str.split("\\s+"));
    Collections.reverse(words);
    String res=String.join(" ", words);
    System.out.print(res);
}
}

```

7, 薯队长写了  $n$  篇笔记, 编号从  $1 \sim n$ , 每篇笔记都获得了不少点赞数。薯队长想从中选出一些笔记, 作一个精选集合。挑选的时候有两个规则:

1. 不能出现连续编号的笔记。
  2. 总点赞总数最多
- 如果满足 1, 2 条件有多种方案, 挑选笔记总数最少的那种

答:

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] nums = new int[n];
        for(int i = 0; i < nums.length; i++) {
            nums[i] = sc.nextInt();
        }
        int[] dp = new int[n+1]; // 总点赞数
        dp[1] = nums[0];
        int[] dpN = new int[n+1]; // 挑选笔记数
        dpN[1] = 1;
        for(int i = 2; i <= n; i++) {
            if(dp[i-1] < dp[i-2] + nums[i-1]) { // 选
                dp[i] = dp[i-2] + nums[i-1];
                dpN[i] = dpN[i-2] + 1;
            } else { // 不选
                dp[i] = dp[i-1];
                dpN[i] = dpN[i-1];
            }
        }
        System.out.println(dp[n] + " " + dpN[n]);
    }
}
```

8,