public class Sample {

    public static void main(String[] args) {

        Sample sample = new Sample();

        int[] answers = {1,2,3,4,5};

        int[] toReturn = sample.solution(answers);

        Arrays.stream(toReturn).forEach(System.out::println);

    }

    public int[] solution(int[] answers) {

        // 배열로 받은 정답을 리스트로 전환, 사용편의

        List<Integer> answersList = new ArrayList<>();

        Arrays.stream(answers).forEach((x)->answersList.add(x));

        // 정답으로 제출할 배열을 만들 리스트

        List<Integer> toReturn = new ArrayList<>();

        // 학생1 의 정답 갯수

        int cnt1 = 0;

        for(int i=1; i<=answers.length; i++) {

            if((i%5!=0 ? i%5 : 5) == answersList.get(i-1)) cnt1++;

        }

        // 학생2 의 정답 갯수

        int cnt2 = 0;

        for(int i=1; i<=answers.length; i++) {

            int k=0;

            switch (i%2) {

            case 1:

                k=2; break;

            case 0:

                switch (i%8) {

                case 2:

                    k=1; break;

                case 4:

                    k=3; break;

                case 6:

                    k=4; break;

                case 0:

                    k=5; break;

                }

            }

            if(k == answersList.get(i-1)) cnt2++;

        }

        // 학생3 의 정답 갯수

        int cnt3 = 0;

        for(int i=1; i<=answers.length; i++) {

            int k=0;

            switch (i%10) {

            case 1: case 2:

                k=3; break;

            case 3: case 4:

                k=1; break;

            case 5: case 6:

                k=2; break;

            case 7: case 8:

                k=4; break;

            case 9: case 0:

                k=5; break;

            }

            if(k == answersList.get(i-1)) cnt3++;

        }

        // 순위 판단 및 동점자를 고려한 고득점자 리스트 반환

        if(cnt1==cnt2 && cnt2 == cnt3) {

            toReturn.add(1);

            toReturn.add(2);

            toReturn.add(3);

        } else if(cnt1==cnt2 && cnt2>cnt3) {

            toReturn.add(1);

            toReturn.add(2);

        } else if(cnt2==cnt3 && cnt3>cnt1) {

            toReturn.add(2);

            toReturn.add(3);

        } else if(cnt1==cnt3 && cnt3>cnt2) {

            toReturn.add(1);

            toReturn.add(3);

        } else {

            String c = cnt1>cnt2? (cnt1>cnt3? "cnt1" : "cnt3") : (cnt2>cnt3? "cnt2": "cnt3");

            toReturn.add(Integer.parseInt(c.substring(c.length()-1)));

        }

        Collections.sort(toReturn);

        // 제출리스트를 배열로 변환

        return toReturn.stream().mapToInt(Integer::valueOf).toArray();

    }

}

**문제 설명**

수포자는 수학을 포기한 사람의 준말입니다. 수포자 삼인방은 모의고사에 수학 문제를 전부 찍으려 합니다. 수포자는 1번 문제부터 마지막 문제까지 다음과 같이 찍습니다.

1번 수포자가 찍는 방식: 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, ...  
2번 수포자가 찍는 방식: 2, 1, 2, 3, 2, 4, 2, 5, 2, 1, 2, 3, 2, 4, 2, 5, ...  
3번 수포자가 찍는 방식: 3, 3, 1, 1, 2, 2, 4, 4, 5, 5, 3, 3, 1, 1, 2, 2, 4, 4, 5, 5, ...

1번 문제부터 마지막 문제까지의 정답이 순서대로 들은 배열 answers가 주어졌을 때, 가장 많은 문제를 맞힌 사람이 누구인지 배열에 담아 return 하도록 solution 함수를 작성해주세요.

제한 조건

* 시험은 최대 10,000 문제로 구성되어있습니다.
* 문제의 정답은 1, 2, 3, 4, 5중 하나입니다.
* 가장 높은 점수를 받은 사람이 여럿일 경우, return하는 값을 오름차순 정렬해주세요.

입출력 예

| **answers** | **return** |
| --- | --- |
| [1,2,3,4,5] | [1] |
| [1,3,2,4,2] | [1,2,3] |

입출력 예 설명

입출력 예 #1

* 수포자 1은 모든 문제를 맞혔습니다.
* 수포자 2는 모든 문제를 틀렸습니다.
* 수포자 3은 모든 문제를 틀렸습니다.

따라서 가장 문제를 많이 맞힌 사람은 수포자 1입니다.

입출력 예 #2

* 모든 사람이 2문제씩을 맞췄습니다.