PC 甲子園 プログラミング部門 予選問題解説

2-A 吉井智明

0. 初めに

どうも、音展前日の深夜（なんなら日付変わってるので当日）の吉井です。眠いです。この番外編部誌では、本編では紹介しきれなかった予選問題の解説をしていきたいと思います。予選本番での僕らの思考順序、後日再考して見つけたポイントなどをできる限りまとめていきます。なお僕らが正解した問題、または後日の再考で答えが導き出せた問題のみの解説になっています。予選問題は最終ページに url を貼り付けておきましたので、そちらからご覧ください。

1. 問題 1

まずは僕の回答です。

```java
import java.util.*;

class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int y = sc.nextInt();
        int ans = 0;
        ans = y - 2002;
        System.out.println(ans);
        sc.close();
    }
}
```

何も難しいことはありません。標準入力で受け取った値から 2002 を引いただけです。

```java
import java.util.Scanner;

public class PC202201 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int Y = sc.nextInt();
        System.out.println(Y - 2002);
        sc.close();
    }
}
```

この田中の回答も同じく受け取った値から 2002 を引いて出力しています。

2. 問題 2

```java
import java.util.*;

class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int d1 = sc.nextInt();
        int m1 = sc.nextInt();
        int d2 = sc.nextInt();
        int m2 = sc.nextInt();
        int today = (d1 * 10) + m1;
        int every = (d2 * 10) + m2;
        if (today >= 375){
            System.out.println(2);
        }
        else{
            if(today >= every){
                System.out.println(1);
            }
            else{
                System.out.println(0);
            }
        }
        sc.close();
    }
}
```

僕のコードです。測定体温と平熱のそれぞれを分単位に変換し、37 度 5 分（即ち 375 分）を加えた 3 つの比較を条件分岐で判別し、各々回答を出力しました。

3. 問題 3

ぱっと見よくあるじゃんけん問題ですが、少し特殊なルールが追加されているおかげで、前問と比べると難易度が上がっています。

```java
import java.util.Scanner;

public class PC202203 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int count = 0;
        for (int i = 0; i < 3; i++) {
            int x = sc.nextInt();
            int y = sc.nextInt();
            if (x != y) {
```

```
            if (x == 0) {
                if (y == 1) {
                    count += 1;
                } else if (y == 2) {
                    count -= 1;
                }
            } else if (x == 1) {
                if (y == 2) {
                    count += 1;
                } else if (y == 0) {
                    count -= 1;
                }
            } else if (x == 2) {
                if (y == 0) {
                    count += 1;
                } else if (y == 1) {
                    count -= 1;
                }
            }
        }
    }

    if (count == 0) {
        System.out.println(-1);
    } else if (count > 0) {
        System.out.println(0);
    } else if (count < 0) {
        System.out.println(1);
    }
    sc.close();
  }
}
```

田中の回答です。

次の手は count によって判断することにしました。3 回勝負をするので 8 行目から for 文を回します。x をアイズ君の手、y をワカマツ君の手とします。12 行目から 32 行目で x が y に勝っているか負けているかを判別し、勝っていたなら count を+1、負けていたなら count を-1 します。あいこの時は count を増減する必要がないので考える必要がありません。よって 12 行目からの条件分岐から外しました。35 行目から 41 行目で count に応じて出力（ count が 0 以上の時アイズ君は現時点で勝っているので「負けるが勝ち」を宣言しないから 0 を出力、count が 0 の時「負けるが勝ち」を宣言しても勝てないので-1 を出力、count が 0 以下の時アイズ君は現時点で負けて

いるので「負けるが勝ち」を宣言するから 1 を出力）しています。

4. 問題 4

まるで数学の場合の数のような問題です。また、この問題以降は予選本番で正解していませんので、予選終了後手を加えた、恐らく正解であろうと思われるコードを掲載していきます。

```java
import java.util.*;

class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int[] hands = new int[N];
        for (int i=0; i<N; i++){
            hands[i] = sc.nextInt();
        }
        //hands に何個 0 があるか確認
        int ans = 0;
        for (int j=0; j<N; j++){
            if (hands[j] == 0){
                ans = ans + 1;
            }
        }
        if (ans == 0){
            ans++;
        }
        System.out.println(ans);
        sc.close();
    }
}
```

僕のコードです。まず問題を解く方針ですが、各人の状態（0 なのか 1 なのか）のうち、0 である人の数が直接答えのグループ数になることを利用します（ただし例外があるので後ほど調整します）。13 行目からの for 文で 0 の個数をカウントしていきます。しかしこの方法のまま答えを出力してしまうと、全員 1（つまり全員が手をつないでいる状態）だった場合に答えが「0 グループ」と出力されてしまいます。しかし勿論この場合の答えは「1 グループ」ですので、20 行目の if 文で特殊例の答えを補正しています。

5. 問題 5

面白い問題ですね。与えられた実験結果を利用して、未知の実験結果を予測する、という問題です。応用すれば実用性がありそうですね。

田中のコードです。

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class PC202205 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        ArrayList<Integer> less = new ArrayList<Integer>();
        ArrayList<Integer> good = new ArrayList<Integer>();
        ArrayList<Integer> much = new ArrayList<Integer>();
        int lessLast; //less の最大
        int good1; //good の最小
        int goodLast; //good の最大
        int much1; //much の最小
        for (int i = 0; i < n; i++) {
            int a = sc.nextInt();
            int s = sc.nextInt();
            if (s == 0) {
                less.add(a);
            } else if (s == 1) {
                good.add(a);
            } else if (s == 2) {
                much.add(a);
            }
        }
        Collections.sort(less);
        Collections.sort(good);
        Collections.sort(much);
        if (!(less.isEmpty() || good.isEmpty() || much.isEmpty())) { ///less good much が全て空でない(全て存在する)とき
            lessLast = less.get(less.size()-1);
            good1 = good.get(0);
            goodLast = good.get(good.size()-1);
            much1 = much.get(0);
            ///処理
            int m = sc.nextInt();
            for (int i = 0; i < m; i++) {
                int b = sc.nextInt();
                if (b<= lessLast) {
```

```
                System.out.println(0);
            } else if (good1 <= b && b <= goodLast) {
                System.out.println(1);
            } else if (much1 <= b) {
                System.out.println(2);
            } else {
                System.out.println(-1);
            }
        }
    } else if (less.isEmpty() && !(good.isEmpty()) && !(much.isEmpty())) { //less が存在しない時
        good1 = good.get(0);
        goodLast = good.get(good.size()-1);
        much1 = much.get(0);
        int m = sc.nextInt();
        for (int i = 0; i < m; i++) {
            int b = sc.nextInt();
            if (good1 <= b && b <= goodLast) {
                System.out.println(1);
            } else if (much1 <= b) {
                System.out.println(2);
            } else {
                System.out.println(-1);
            }
        }
    } else if (!(less.isEmpty()) && good.isEmpty() && !(much.isEmpty())) { //good が存在しない時
        lessLast = less.get(less.size()-1);
        much1 = much.get(0);
        int m = sc.nextInt();
        for (int i = 0; i < m; i++) {
            int b = sc.nextInt();
            if (b<= lessLast) {
                System.out.println(0);
            } else if (much1 <= b) {
                System.out.println(2);
            } else {
                System.out.println(-1);
            }
        }
    } else if (!(less.isEmpty()) && !(good.isEmpty()) && much.isEmpty()) { //much が存在しない時
        lessLast = less.get(less.size()-1);
        good1 = good.get(0);
        goodLast = good.get(good.size()-1);
        int m = sc.nextInt();
        for (int i = 0; i < m; i++) {
            int b = sc.nextInt();
            if (b<= lessLast) {
                System.out.println(0);
            } else if (good1 <= b && b <= goodLast) {
                System.out.println(1);
```

```
                } else {
                    System.out.println(-1);
                }
            }
        } else if (less.isEmpty() && good.isEmpty() && !(much.isEmpty())) { //less good が存在しない時
            much1 = much.get(0);
            int m = sc.nextInt();
            for (int i = 0; i < m; i++) {
                int b = sc.nextInt();
                if (much1 <= b) {
                    System.out.println(2);
                } else {
                    System.out.println(-1);
                }
            }
        } else if (!(less.isEmpty()) && good.isEmpty() && much.isEmpty()) { //much good が存在しない時
            lessLast = less.get(less.size()-1);
            int m = sc.nextInt();
            for (int i = 0; i < m; i++) {
                int b = sc.nextInt();
                if (b<= lessLast) {
                    System.out.println(0);
                } else {
                    System.out.println(-1);
                }
            }
        } else if (much.isEmpty() && less.isEmpty() && !(good.isEmpty())) { //less much が存在しない時
            good1 = good.get(0);
            goodLast = good.get(good.size()-1);
            int m = sc.nextInt();
            for (int i = 0; i < m; i++) {
                int b = sc.nextInt();
                if (good1 <= b && b <= goodLast) {
                    System.out.println(1);
                } else {
                    System.out.println(-1);
                }
            }
        }
        sc.close();
    }
}
```

以降茹で足りないを less、ちょうどいいを good、茹で過ぎを much

lessLast は less の最大、good1 は good の最小

goodLast は good の最大、much1 は much の最小　とします

18 行目から 28 行目で、標準入力で受け取った時間のデータを less、good、much のリストに分けて入れます。30 行目から 32 行目でそのリストを大きくない順（本校某数学教師 K 先生リスペクト）にソートしています。

ここで鬼の条件分岐で判定をしていきます。

34 行目から 52 行目で less good much が全て空でない（全て存在する）時

52 行目から 68 行目で less が存在しない時

69 行目から 82 行目で good が存在しない時

83 行目から 97 行目で much が存在しない時

98 行目から 108 行目で less good が存在しない時

109 行目から 119 行目で much good が存在しない時

120 行目から 132 行目で less much が存在しない時

この 7 通りの場合分けで、b_i 分茹でた時の具合を出力しました。

6.終わりに

僕らが答えを導き出すことができたのは以上の問題までででした。本当はもう少し考える時間があれば良かったのですが、~~音展当日の早朝に書いているので~~、以上とします。

また、今回の部誌番外編に掲載したコードには、多くのミス等があると思われます。なので、PC 部さんが開発した 2ch 風 web サイトに専用スレッドを立ち上げておきました。何かしらのご指摘や質問などはそちらに書き込んでいただければ、僕らが可能な限り回答させていただきたいと思っています。よろしくお願いいたします。

7.url 集

パソコン甲子園 2022 プログラミング部門 予選問題 https://bit.ly/3xLrS1H

学年 1 位の 2 ちゃんねる 物理部 質問スレッド https://bit.ly/3C1Sf5R