```java
//Tushaar Jain
//April 9th, 2020
//Int 5
//Enloe HS
//Contest #4
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class Main {
  static ArrayList<Integer> opponent = new ArrayList<Integer>();
  static ArrayList<Integer> me = new ArrayList<Integer>();
  static ArrayList<Integer> rolls = new ArrayList<Integer>();

  private static int getIndexOfLowest() {
    int lowest = Integer.MAX_VALUE;
    int index = 0;
    for (int i = 0; i < me.size(); i++) {
      if (me.get(i) < lowest) {
        lowest = me.get(i);
        index = i;
      }
    }
    return index;
  }

  private static boolean isPrime(int n) {
    for (int i = 2; i <= (int) Math.sqrt(n); i++) {
      if (n % i == 0)
        return false;
    }
    return true;
  }

  private static Boolean isOccupied(int n) {
    for (int i : me) {
      if (i == n) {
        return true;
      }
    }
    for (int i : opponent) {
      if (i == n) {
        return true;
```

```
    }
  }
  return false;
}

private static void move(int index, int roll) {
  int prevpos = me.get(index);
  int newpos = prevpos;

  if (isOccupied(prevpos + roll)) {
    return;
  } else if (!isOccupied(prevpos + roll)) {
    newpos = prevpos + roll;
    me.set(index, newpos);
  }

  if (newpos == 52) {
    me.remove(index);
    return;
  }
  if (newpos > 52) {
    newpos = prevpos;
    me.set(index, newpos);
    return;
  }

  if (isPrime(newpos)) {
    for (int i = 1; i <= 6; i++) {
      if (isOccupied(newpos + i)) {
        newpos = newpos + i - 1;
        me.set(index, newpos);
        return;
      }
    }
    newpos = newpos + 6;
    me.set(index, newpos);
    return;
  }

  if (Math.sqrt(newpos) == Math.floor(Math.sqrt(newpos)) && newpos > 4) {
    for (int i = 1; i <= 6; i++) {
      if (isOccupied(newpos - i)) {
        newpos = newpos - i + 1;
        me.set(index, newpos);
        return;
```

```
      }
    }
    newpos = newpos - 6;
    me.set(index, newpos);
    return;
  }
  if (madeCornerMove(prevpos, roll, newpos)) {
    me.set(index, prevpos);
    for (int i = 1; i <= roll; i++) {
      if ((prevpos + i) % roll == 0 && !isOccupied(prevpos + i)) {
        newpos = prevpos + i;
        me.set(index, newpos);
        return;
      }
    }
    newpos = prevpos;
    me.set(index, newpos);
    return;
  }
}

private static boolean madeCornerMove(int prevpos, int roll, int newpos) {
  int[] corners = { 7, 12, 17, 22, 27, 35, 40, 45, 50 };
  for (int i : corners) {
    if (prevpos < i && newpos > i) {
      return true;
    }
  }
  return false;
}

public static void main(String args[]) throws FileNotFoundException {
  File myfile = new File("int.txt");
  Scanner myscanner = new Scanner(myfile);
  for (int i = 0; i < 5; i++) {
    me.clear();
    opponent.clear();
    rolls.clear();
    String inString = myscanner.nextLine().trim();
    String[] splited = inString.split("\\s+");
    int count = 0;
    for (String n : splited) {
      if (count < 3) {
        opponent.add(Integer.parseInt(n));
      } else if (count < 6) {
```

```
            me.add(Integer.parseInt(n));
          } else if (count != 6) {
            rolls.add(Integer.parseInt(n));
          }
          count++;
        }
        for (int j : rolls) {
          int index = getIndexOfLowest();
          move(index, j);
        }
        if(!me.isEmpty()){
        Collections.sort(me);
        for (int j : me) {
          System.out.print(j + " ");
        }}
        else{
          System.out.print("GAMEOVER");
        }
        System.out.println();
      }
    }
}
```