```
//AnjaliKesari
//Intermediate -Patolli

import java.io.*;
import java.util.*;
public class AnjaliKesari_INTc4Patolli{
   public static void main(String[] args) throws Exception {
      File file = new File("input.txt");
      Scanner sc = new Scanner(file);
      while (sc.hasNextLine()){
         String a="";
         a=sc.nextLine();
         game(a);
      }
   }
   public static void game(String a){
      int[] grid = new int[52];
      for(int i=1; i<=52; i++){
         grid[i-1]=i;}
      String[] in = a.split(" ");
      for(int i=0; i<=2; i++){//opponent
         grid[Integer.parseInt(in[i])-1]=99;}
      for(int i=3; i<=5; i++){//me
         grid[Integer.parseInt(in[i])-1]=0;}
      int[] dice = new int[Integer.parseInt(in[6])];//dice
      for(int i =0; i<dice.length; i++){
         dice[i]=Integer.parseInt(in[i+7]);
      }
   //       System.out.println("lowest index: "+findLowest(grid));
   //       System.out.println("Dice rolls: "+Arrays.toString(dice));
   //       System.out.println("original: "+Arrays.toString(grid));
   //       System.out.println("Markers: "+findAll(grid));

      for(int i=0; i<dice.length; i++){
         int add=0;
         int before=findLowest(grid);
         add=findLowest(grid)+dice[i]-1;//adding dice
         if(add<52){
            if((grid[add]!=0)&&(grid[add]!=99)){
               if(grid[add]==52){
                  grid[add]=52;
               }
               else{
                  grid[add]=0;
                  grid[findLowest(grid)-1]=findLowest(grid);
                  if(grid[51]==0){
                     grid[51]=52;}
               }
               if(isPrime(add+1)){
                  prime(add,grid);
               }
               else if(isSquare(add+1)){
                  square(add,grid);
               }
               else if(isNine(before,dice[i])){
                  grid[add]=add+1;
                  grid[before-1]=0;
                  nine(before, dice[i], grid);
               }
            }
         }
         else{
            grid[findLowest(grid)-1]=0;//skips turn
         }
      }
      //System.out.println("final: "+Arrays.toString(grid));
      System.out.println(findAll(grid));
   }
   public static int findLowest(int [] grid){
```

```
   int min=0;//lowest marker position
   for(int i=0; i<grid.length; i++){
      if(grid[i]==0){
         min=i+1;
         break;
      }
   }
   return min;
}
public static String findAll(int [] grid){//markers positions
   String ans="";
   for(int i=0; i<grid.length; i++){
      if(grid[i]==0){
         ans+=(i+1)+" ";
      }
   }
   return ans;
}
public static void prime(int add, int [] grid){
   boolean b=false; int stop=0;
   for(int i=add+1; i<=add+6; i++){
      if((grid[i]!=0)&&(grid[i]!=99)){
         b=true;
      }
      else{
         b=false;
         stop=i;
         break;
      }
   }
   if(b){
      grid[add]=add+1;
      grid[add+6]=0;
   }
   else{
      grid[add]=add+1;
      grid[stop-1]=0;
   }
}
public static boolean isPrime(int num){
   boolean b = false; boolean prime=false;
   for(int i = 2; i <= num/2; ++i){
      if(num % i == 0){
         b = true;
      }}
   if (!b){
      prime=true;}
   else{
      prime=false;}
   return prime;
}
public static boolean isSquare(int num){
   boolean b = false; boolean square=false;
   for(int i = 1; i <= num/2; i++){
      if((double)num / i == (double)i){
         b = true;
         break;
      }}
   if (b){
      square=true;}
   else{
      square=false;}
   return square;
}
public static void square(int add, int [] grid){
   boolean b=false; int stop=0;
   for(int i=add-1; i>=add-6; i--){
      if((grid[i]!=0)&&(grid[i]!=99)){
         b=true;
```

```
         }
         else{
            b=false;
            stop=i;
            break;
         }
      }
      if(b){
         grid[add]=add+1;
         grid[add-6]=0;
      }
      else{
         grid[add]=add+1;
         grid[stop+1]=0;
      }
   }
   public static boolean isNine (int bef, int dice){
      boolean n=false; int pos=0;
      int [] arr = {7, 12, 17, 22, 27, 35, 40, 45, 50};
      for(int i=0; i<arr.length; i++){
         if((bef<arr[i])&&((bef+dice)>arr[i])){
            n=true; pos=arr[i];
            break;
         }
      }
      return n;
   }
   public static void nine (int bef, int dice, int [] grid){
      int end=0; boolean b=false;
      for(int i=bef; i<bef+dice; i++){
         if((grid[i]!=0)&&(grid[i]!=99)){
            if((grid[i]%dice==0)){
               b=true; end=i;
               break;
            }
         }
      }
      if(b){
         grid[bef-1]=bef;
         grid[end]=0;
         if(end!=bef-1+dice){
            grid[bef-1+dice]=bef+dice;
         }
      }
   }
}
```