Write the following methods that *return a lambda expression* performing a specified action:

1. PerformOperation isOdd(): The lambda expression must return if a number is odd or if it is even.
2. PerformOperation isPrime(): The lambda expression must return if a number is prime or if it is composite.
3. PerformOperation isPalindrome(): The lambda expression must return if a number is a palindrome or if it is not.

Input Format

Input is handled for you by the locked stub code in your editor.

Output Format

The locked stub code in your editor will print lines of output.

Sample Input

The first line contains an integer, (the number of test cases).

The subsequent lines each describe a test case in the form of space-separated integers:

The first integer specifies the condition to check for (for Odd/Even, for Prime, or for Palindrome). The second integer denotes the number to be checked.

```
import java.io.*;
import java.util.*;
interface PerformOperation {
 boolean check(int a);
}
class MyMath {
 public static boolean checker(PerformOperation p, int num) {
  return p.check(num);
 }

   // Write your code here
   public static PerformOperation isOdd() {
    return (int a) -> a % 2 != 0;
   }
    // Lambda for prime check
```

```
     public static PerformOperation isPrime() {
        return (int a) -> { if (a <= 1) return false;
         if (a == 2) return true;
          if (a % 2 == 0) return false;
          for (int i = 3; i <= Math.sqrt(a); i += 2) {
             if (a % i == 0) return false;
             } return true;
             };
              }
    // Lambda for palindrome check
     public static PerformOperation isPalindrome() { return (int
a) -> { String s = String.valueOf(a); return new
StringBuilder(s).reverse().toString().equals(s);
      }; }

}

public class Solution {

 public static void main(String[] args) throws IOException {
  MyMath ob = new MyMath();
  BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
  int T = Integer.parseInt(br.readLine());
  PerformOperation op;
  boolean ret = false;
  String ans = null;
  while (T--> 0) {
   String s = br.readLine().trim();
   StringTokenizer st = new StringTokenizer(s);
   int ch = Integer.parseInt(st.nextToken());
   int num = Integer.parseInt(st.nextToken());
   if (ch == 1) {
    op = ob.isOdd();
    ret = ob.checker(op, num);
    ans = (ret) ? "ODD" : "EVEN";
   } else if (ch == 2) {
    op = ob.isPrime();
    ret = ob.checker(op, num);
    ans = (ret) ? "PRIME" : "COMPOSITE";
   } else if (ch == 3) {
    op = ob.isPalindrome();
    ret = ob.checker(op, num);
    ans = (ret) ? "PALINDROME" : "NOT PALINDROME";

   }
```

```
    System.out.println(ans);
   }
  }
}
```