关于下面代码说法正确的是：

```
public class Demo {
  private Demo() {}
  private static class Singleton {
    private static final Demo INSTANCE = new Demo();
  }
  public static Demo getInstance() {
    return Singleton.INSTANCE;
  }
}
```

正确答案: A　你的答案: 空 (错误)
线程安全，懒加载
线程安全，启动加载
非线程安全，懒加载
非线程安全，启动加载

完全二叉树是指深度为 K 的，有 n 个结点的二叉树，当且仅当其每一个结点都与深度为 K 的满二叉树中编号从 1 至 n 的结点一 一对应将一棵有 50 个结点的完全二叉树按节点编号，如根节点的编号为 1，那么编号为 25 的结点是（ ）？
正确答案: B　你的答案: 空 (错误)
无左、右孩子
有左孩子，无右孩子
有右孩子，无左孩子
有左、右孩子
对象的浅拷贝和深拷贝区别是什么？在 JAVA 中如何实现？
浅拷贝：“被复制对像”基本类型的值和对象的引用。换言之，浅复制仅仅复制所考虑的对象，而不复制它所引用的对象。
深拷贝：“被复制对象”基本类型的值，对象的引用将重新指向一个被复制过的新对象。换言之，深复制把要复制的对象所引用的对象都复制了一遍。
使用 Thread 类和 Runnable 方法来创建一个线程的区别是什么？
通常的理解是继承 Thread 类或实现 Runnable 接口，重写 run 方法，然后用 new Thread(Runnable run).start()实现。
实际常用 lambda 表达式实现：new Thread(() ->{ //doSomething }).start() 。
当然线程的创建和销毁都是需要额外的开销的，所以项目中常用线程池代替。可以自主创建 ThreadPoolExecuter 或者使用 Executers 类中的实现。

定义$mathitS(n)$，表示n在十进制下的各位数字和。

现在给定一个n,请你求出最小正整数x，满足x.

```
import java.util.Scanner;

public class Main{
```

```
    public static void main(String args[]){
        Scanner in = new Scanner(System.in);
        int length = in.nextInt();
        for(int i=0;i<length;i++){
            System.out.println(s(in.nextInt()));
        }
    }

    public static String s(int x){
        StringBuilder builder = new StringBuilder();
        while(x-9>0){
            builder.append("9");
            x-=9;
        }
        String result = String.valueOf(x)+builder.toString();
        return result;
    }
}
```

小易给定你数字和系数。每次操作你可以将变成或者将变成。问至少几次操作使得。

```
import java.util.*;

public class Main {

    public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        int n=input.nextInt();
        input.nextLine();
        for(int i=0;i<n;i++){
            String[] s = input.nextLine().split(" ");
            int A=Integer.parseInt(s[0]);
            int B=Integer.parseInt(s[1]);
            int p=Integer.parseInt(s[2]);
            int q=Integer.parseInt(s[3]);
            System.out.println(count(A,B,p,q,0));
        }
    }

    private static int count(double A,double B,double p,double q,int num){
        if(B-A<=p)
            return num+1;
        else if((B-A)/p<=q)
            return num+2;
```

```
        else {
            return count(A,B,p*q*q,q,num+2);
        }
    }

}
```