

Zadanie 7 (8p)

Napisać w Pythonie program implementujący test pierwszości Millera-Rabina.

```
public boolean Miller_Rabbin_isPrime ( Integer s , BigInteger r , Long t ){
    BigInteger n = BigInteger.TWO.pow( s ).multiply( r ).add( BigInteger.ONE );
    return Miller_Rabbin_isPrime( BigInteger.valueOf(s), r , t , n );
}

public boolean Miller_Rabbin_isPrime ( BigInteger s , BigInteger r , Long t , BigInteger n ){
    for ( Long i=0L; i<t ; i++){
        BigInteger a;
        BigInteger j;
        BigInteger y;
        do {
            a = generatorBBC(10).get(0).add(BigInteger.TWO);
        } while ( a.compareTo( n ) < 0 );
        BigInteger m=( n.subtract( BigInteger.ONE ).divide( BigInteger.TWO ) );
        j=s;
        while( j.toString().indexOf(".")>0 ){
            y = a.modPow( m, n ).mod( n );
            if ( y.mod(n).compareTo( BigInteger.ONE )!=1L && y.compareTo( ( n.subtract(BigInteger.ONE)).mod(n) )!=0 ){
                System.out.println("n jest liczba złożoną");
                return false;
            }
            if ( y.compareTo( ( n.subtract(BigInteger.ONE)).mod(n) )==0 ){
                j=BigInteger.ZERO;
            } else {
                j=j.subtract(BigInteger.ONE);
                m=m.divide(BigInteger.TWO);
            }
        }
    }
    System.out.println("n jest liczba pierwszą");
    return true;
}
```

n jest liczba pierwszą: 2097153
n jest liczba złożoną: 1917273370

