17/12/2020 ASHWIN KUMAR. J 312217104020

Aim:

To Develop a java program to implement the miller rabin primality checking to check primality of a given number.

Procedure:

1. Handle base Test cases till value 3.
2. When n is even, return false.
3. Find odd number d, where $n-1 = d * 2^r$.
   where $r > 0$ and n is odd and n-1 is even.
4. Do miller rabin Primality Test:
   1. Choose random 'a' between 2 and n-2.
   2. compute $x = (a \wedge d) \% n$
   3. When $x = 1$ or $n-1$, return True.
   4. Else repeat steps until d!=(n-1):
      - .1 $x = (x * x) \% n$
      - .2 Now when x becomes 1, return false.
      - .3 else return True when $x = (n-1)$.

Code :

```
import java.io.*;
import java.math.*;
import java.util.Scanner;

class Main {
 static int power(int a,int d,int n){
  int res = 1;
  a = a % n;

  while(d>0){
   if(d%2!=0)
     res = (res*a) % n;

   d=d/2;
   a = (a*a) % n;
  }
  return res;
}

 static boolean miller(int n, int d){
  int a = 2 + (int) (Math.random() % (n-4));

  int x = power(a,d,n);

  if(x==1 || x==n-1)
   return true;

  while(d!=n-1){
```

```
    x = (x*x)%n;
    d*=2;

    if(x==1)
      return false;
    if(x==n-1)
      return true;
  }
  return false;
}

static boolean prime(int n,int k){
  if(n<=1 || n==4)
    return false;
  if(n<=3)
    return true;

  int d = n-1;

  while(d%2 == 0)
    d/=2;
  for(int i=0;i<k;i++){
    if(miller(n,d))
      return true;
  }
  return false;
}

public static void main(String[] args) {
```

```
  int k = 3;
  int n = 0;

  Scanner sc = new Scanner(System.in);
  System.out.print("Number : ");
  n = sc.nextInt();

  if(prime(n,k))
    System.out.println(n + " is Prime");
  else
    System.out.println(n + " is Composite");
 }
}
```

Output :