**Experiment number 1 b**

**Aim:** A Mersenne prime is a prime number that has the form 2p−1 where p is a positive number greater than 1. Write a program that calculates candidate Mersenne primes 2p − 1 for 2≤p≤31 using multiple classes and objects. Then test the number to see if it is prime. If you detect that the number is prime, print out the number and the value of p.

**Theory:** A Mersenne prime is a prime number that has the form 2p−1 where p is a positive number greater than 1.

**Algorithm:**

1. public class mersenne
2. public static void main
3. create an object u of checkPrime and an object v of Print
4. for(int p=2;p<=31;p++){ double l=2^p-1
5. int x=u.prime(l) and v.print(x,l)
6. inside class checkPrime, theres a function prime(double a)
7. for(int i=2 ; i<=sqrt of a ; i++) and if(a%i==0), then return 0, else continue
8. if the loop runs completely, then return 1 which means its a prime number
9. inside class Print, there a function prime(int x, double l){ if x==1, then l is a mersenne number and print it.}

**Code:**

import java.util.\*;

import java.lang.\*;

public class mersenne{

public static void main(String args[]){

checkPrime u= new checkPrime();

Print v= new Print();

for(int p=2;p<=31;p++){

double l=(double)Math.pow(2,p)-1;

int x=u.prime(l);

v.print(x,l);

}

}

}

class checkPrime{

public static int prime(double a){

int r=(int)Math.sqrt(a);

for(int i=2;i<=r;i++){

if(a%i==0){

return 0;

}

}

return 1;

}

}

class Print{

public static void print(int x, double l){

if(x==1){

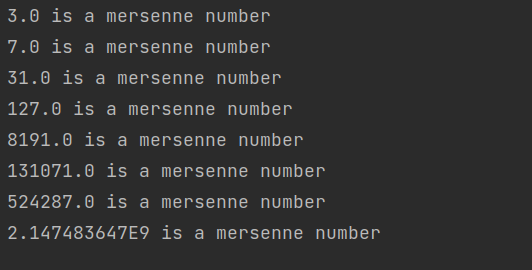
System.out.println(l+" is a mersenne number");

}

}

}

**Output:**

****

**Conclusion:**

By writing this code, I learnt how I could use different classes and functions within them to perform different operations. Here, I created one class for taking an input, another class for checking whether the number is prime or not, and a third class just for printing.