Java Lab

Aadhitya Swarnesh

[redacted]

4 February 2020

Q1)

```
class Divisors {
    long[] findDivisors(long a, long b) {
        long[] ans = { 0, 0 };
        long i, j;
        int d;
        for (i = a; i <= b; i++) {
            d = 1;
            for (j = 1; j <= (i/2); j++) {
                if (i % j == 0) {
                    d++;
                }
            }
            if (d > ans[1]) {
                ans[0] = i;
                ans[1] = d;
            }
        }
        return (ans);
    }
}

class NewThread extends Thread {
    Divisors ob;
    long num, div, a, b;

    NewThread(long a, long b, Divisors obj) {
        super();
        this.ob = obj;
        num = 0;
        div = 0;
        this.a = a;
        this.b = b;
        start();
    }

    public void run() {
        long[] ans = new long[2];
        ans = ob.findDivisors(a, b);
        this.num = ans[0];
        this.div = ans[1];
    }

    long[] getAnswer() {
        long[] ans = new long[2];
        ans[0] = num;
        ans[1] = div;
        return (ans);
```

```
    }
}

class q1 {
    public static void main(String[] args) {
        Divisors ob = new Divisors();
        long num = 0, div = 0, p = 1, q = 10000;
        long[] ans = new long[2];
        NewThread nobj;
        while (q <= 100000) {
            nobj = new NewThread(p, q, ob);
            try {
                //nobj.join();
                if (Thread.currentThread().getName().equals("main")) {
                    Thread.sleep(6000);
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            ans = nobj.getAnswer();
            if (ans[1] > div) {
                num = ans[0];
                div = ans[1];
            }
            p += 10000;
            q += 10000;
        }
        System.out.println("The number with maximum divisors : " + num + " and it has " + div + "
divisors.");
    }
}
```

```
(base) Aadhityas-MacBook-Air:4Feb2020 aadhitya$ javac q1.java
(base) Aadhityas-MacBook-Air:4Feb2020 aadhitya$ java q1
The number with maximum divisors : 83160 and it has 128 divisors.
(base) Aadhityas-MacBook-Air:4Feb2020 aadhitya$
```

Q2)

```
import java.util.*;
class q2
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int num, i, j;
        System.out.println("Enter the number : ");
        num = sc.nextInt();
        boolean[] isPrime = new boolean[num+1];
        Arrays.fill(isPrime, Boolean.TRUE);
        isPrime[0] = false;
        isPrime[1] = false;
        for(i=2;i*i<num;i++)
        {
            if(isPrime[i] == true)
            {
                j = 2;
                while(i*j <= num)
                {
                    isPrime[i*j] = false;
                    j++;
                }
            }
        }
        System.out.println("The prime numbers are : ");
        for(i=0;i<=num;i++)
        {
            if(isPrime[i] == true)
            {
                System.out.println(i);
            }
        }
        sc.close();
    }
}
```

```
(base) Aadhityas-MacBook-Air:4Feb2020 aadhitya$ javac q2.java
(base) Aadhityas-MacBook-Air:4Feb2020 aadhitya$ java q2
Enter the number :
100
The prime numbers are :
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
(base) Aadhityas-MacBook-Air:4Feb2020 aadhitya$
```

Q3)

```
class Queues
{
    int n;
    boolean valueSet = false;
    synchronized int get()
    {
        while(!valueSet)
        {
            try
            {
                wait();
            }catch(InterruptedException e)
            {
                System.out.println("InterruptedException caught");
            }
        }
        System.out.println("Got: " + n);
        this.valueSet = false;
        notify();
        return n;
    }
    synchronized void put(int n)
    {
        while(valueSet)
        {
            try
            {
                wait();
            } catch(InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        }
        this.n = n;
        valueSet = true;
        System.out.println("Put: " + n);
        notify();
    }
}

class Producer implements Runnable
{
    Queues q;
    Producer(Queues q)
    {
        this.q = q;
        new Thread(this, "Producer").start();
    }
    public void run()
    {
        int i = 0;
        while(i<5)
        {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
```

```
    Queues q;
    Consumer(Queues q)
    {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
    public void run()
    {
        int i = 0;
        while(i<5)
        {
            q.get();
            i++;
        }
    }
}

class q3
{
    public static void main(String args[])
    {
        Queues q = new Queues();
        new Producer(q);
        new Consumer(q);
    }
}
```

```
(base) Aadhityas-MacBook-Air:4Feb2020 aadhitya$ javac q3.java
(base) Aadhityas-MacBook-Air:4Feb2020 aadhitya$ java q3
Put: 0
Got: 0
Put: 1
Got: 1
Put: 2
Got: 2
Put: 3
Got: 3
Put: 4
Got: 4
(base) Aadhityas-MacBook-Air:4Feb2020 aadhitya$
```