LAB7(B.Bhagyasri)

1. **Write a program that creates two threads. Each thread should print its thread ID (TID) and a unique message to the console. Ensure that the output from both threads is interleaved.**

```
package Lab7;


public class InterleavedThread {


	public static void main(String[] args) {
		Thread t1=new Thread(new Interleaved("Thread1"));
		Thread t2=new Thread(new Interleaved("Thread2"));
		t1.start();
		t2.start();
	 }
}


package Lab7;
public class Interleaved implements Runnable{
public String message;
public Interleaved(String msg) {
message=msg;
	}
	public void run() {
	for(int i=0;i<5;i++) {
	System.out.println(Thread.currentThread().getId()+"- "+message);
	try {
		Thread.sleep(100);
	}
		catch(Exception e) {
		e.printStackTrace();
			}
```

```
            }
        }

}
```

Output:

21- Thread1

22- Thread2

22- Thread2

21- Thread1

22- Thread2

21- Thread1

22- Thread2

21- Thread1

22- Thread2

21- Thread1

2. Write a program that creates multiple threads with different priorities. Observe how the operating system schedules threads with different priorities and explain the results.

```
package Lab7;


public class Thread_prior {


        public static void main(String[] args) {
                // TODO Auto-generated method stub
                Threadone t1=new Threadone();
                Threadtwo t2=new Threadtwo();
                Threadthree t3=new Threadthree();
                t1.setPriority(Thread.MIN_PRIORITY);
```

```
                t2.setPriority(Thread.MAX_PRIORITY);
                t3.setPriority(Thread.MAX_PRIORITY);
                t1.start();
                t2.start();
                t3.start();
                }
}
```

```
package Lab7;

public class Threadone extends Thread{


                // TODO Auto-generated method stub
                public void run() {
                for(int i=0;i<6;i++) {
                System.out.println("i :"+i);
                try
                {
                sleep(500);
                                }
                catch(Exception e) {
                System.out.println(e);
                }
                }
                System.out.println("End of one");
        }
    }
```

```
package Lab7;
public class Threadtwo extends Thread{
                public void run() {
                for(int j=0;j<6;j++) {
                System.out.println("j :"+j);
                try
                {
                sleep(500);
                }
                catch(Exception e) {
                System.out.println(e);

                        }
                  }
              }
          }

package Lab7;

public class Threadthree extends Thread{
        public void run() {
                for(int k=0;k<=6;k++) {
                System.out.println("k :"+k);
                try
                {
                sleep(500);
                }
                catch(Exception e) {
                System.out.println(e);
```

```
                    }
                    }
                System.out.println("End of three");
}}
```

Output:

k :0

j :0

i :0

k :1

j :1

i :1

k :2

i :2

j :2

j :3

i :3

k :3

i :4

k :4

j :4

i :5

k :5

j :5

k :6

End of one

End of three

3. Write a Java program that creates two threads and prints "Thread A" from the first thread and "Thread B" from the second thread. Make sure both threads run concurrently.

```
package Lab7;

public class Thread11 extends Thread {
private String message;
public void run(String msg) {
                message=msg;
                System.out.println(msg);
                }
}
```

```
package Lab7;

public class Thread11sim {

        public static void main(String[] args) {
                Thread11 t1=new Thread11();
                System.out.println("Thread A from the first thread");
                Thread11 t2=new Thread11();
                System.out.println("Thread B from the second thread");
                t1.run();
                }
}
```

Output:

```
Thread A from the first thread
Thread B from the second thread
```