```java
@Override
public void run() {
    for(BigInteger i = start; i.compareTo(end) == -1; i = i.add(BigInteger.ONE)) {
    //i를 ()안의 값과 compareTo해서 i가 작으면 -1, 같으면 ==, 크면 1을 출력한다.

        if(i.mod(new BigInteger(String.valueOf(option))).compareTo(BigInteger.ZERO) == 0) {
        //option으로 들어온 값은 int이기 때문에 빅인티저인 i와 비교하려면 new BigInteger를 통해 i와 같은 빅인티저 형태로 만들어주어야한다!
        //mod연산은 빅인티저에서 쓰이는 나머지 연산이다
        //(String.valueOf(option))은 int를 BigInteger로 바꾸기 위해 쓰이는것 같은데 정확히 무슨뜻이지??

            System.out.println(threadIdx + "번째 스레드의 계산, " + option + "의 배수는 " + i);

            localSum = localSum.add(i);
        }
```

```
}
public void calcEachThreadStart() {
    for(int i=0; i<THREAD_NUM; i++) {
        calThr[i].start();
        //.run()은 thread들이 순차적으로 계산이 되지만,
        //.start()는 thread들이 cpu의 지시대로 critical section에 접근하면서 무작위로 계산이 진행된다!
        //.run()과 .start()의 작동이 왜 다른건지 궁금합니다!
    }
}
```