第一题

In [ ]:
```
var sum=0
for(i<- 1 until 100 if i%2 !=0)sum+=i
println(sum)
```

```
2500
```

Out[ ]:
```
sum: Int = 2500
```

则100以内所有奇数的总和为2500

第二题

In [ ]:
```
var sum=0.0  //初值一定要赋0.0
for(i<- 1 to 100)sum+=1.0/i.toDouble
println(sum)
```

```
5.187377517639621
```

Out[ ]:
```
sum: Double = 5.187377517639621
```

则待求结果为5.187377517639621

第三题

In [ ]:
```
def fac(n:Int):Long={  //一定要写Long
    if(n>1) n*fac(n-1)
    else 1   //不能写两个if...
}
fac(15)
```

Out[ ]:
```
fac: (n: Int)Long

res7: Long = 1307674368000
```

则15!=1307674368000

第四题

In [ ]:
```
def feb(n:Int):Long={
    if(n>1) feb(n-1)+feb(n-2)
    else if(n==1) 1
    else 0
}
feb(40)
```

Out[ ]:
```
feb: (n: Int)Long

res8: Long = 102334155
```

则斐波那契数列第四十项的值为102334155

第五题

In [ ]:
```
val rdd=sc.parallelize(Array(1,3,5,7,9,11,13,15,17,19))
val sum=rdd.reduce((x,y)=>x+y)
val rdd1=rdd.map(x=>x-sum.toDouble/10.0)
val rdd2=rdd1.map(x=>x*x)
val sum1=rdd2.reduce((x,y)=>x+y)
val out=sum1.toDouble/9.0
```

Out[ ]:
```
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[30] at parallelize at <
console>:36

sum: Int = 100

rdd1: org.apache.spark.rdd.RDD[Double] = MapPartitionsRDD[31] at map at <console>:
38

rdd2: org.apache.spark.rdd.RDD[Double] = MapPartitionsRDD[32] at map at <console>:
39

sum1: Double = 330.0

out: Double = 36.666666666666664
```

则原数据的样本方差为36.666666666666664

第六题

In [ ]:
```
import breeze.linalg._
def cf(x:DenseVector[Int],k:Int):DenseVector[Int]={  //DenseVector一定要写中括号及元
    if(k==0) x
    else {
        val x1=cf(x,k-1)
        x1(1 to x1.length-1)-x1(0 to x1.length-2)
    }
}
val x=DenseVector(Array(1,5,3,7,4,2,7,8,2,4,6,9,3,3,76,8))
cf(x,5)
```

Out[ ]:
```
import breeze.linalg._

cf: (x: breeze.linalg.DenseVector[Int], k: Int)breeze.linalg.DenseVector[Int]

x: breeze.linalg.DenseVector[Int] = DenseVector(1, 5, 3, 7, 4, 2, 7, 8, 2, 4, 6,
9, 3, 3, 76, 8)

res11: breeze.linalg.DenseVector[Int] = DenseVector(46, -23, -15, 25, 10, -41, 32,
-20, 36, 27, -333)
```

则题中向量的5阶差分为(46, -23, -15, 25, 10, -41, 32, -20, 36, 27, -333)