Course ID: CS 501
Name: Mengmeng Xue Student ID: 19519
Hw2

Description:
http://npu85.npu.edu/~henry/npu/classes/algorithm/recursive/slide/exercise_recursive.html
Q2 ==> Summation of N odd numbers

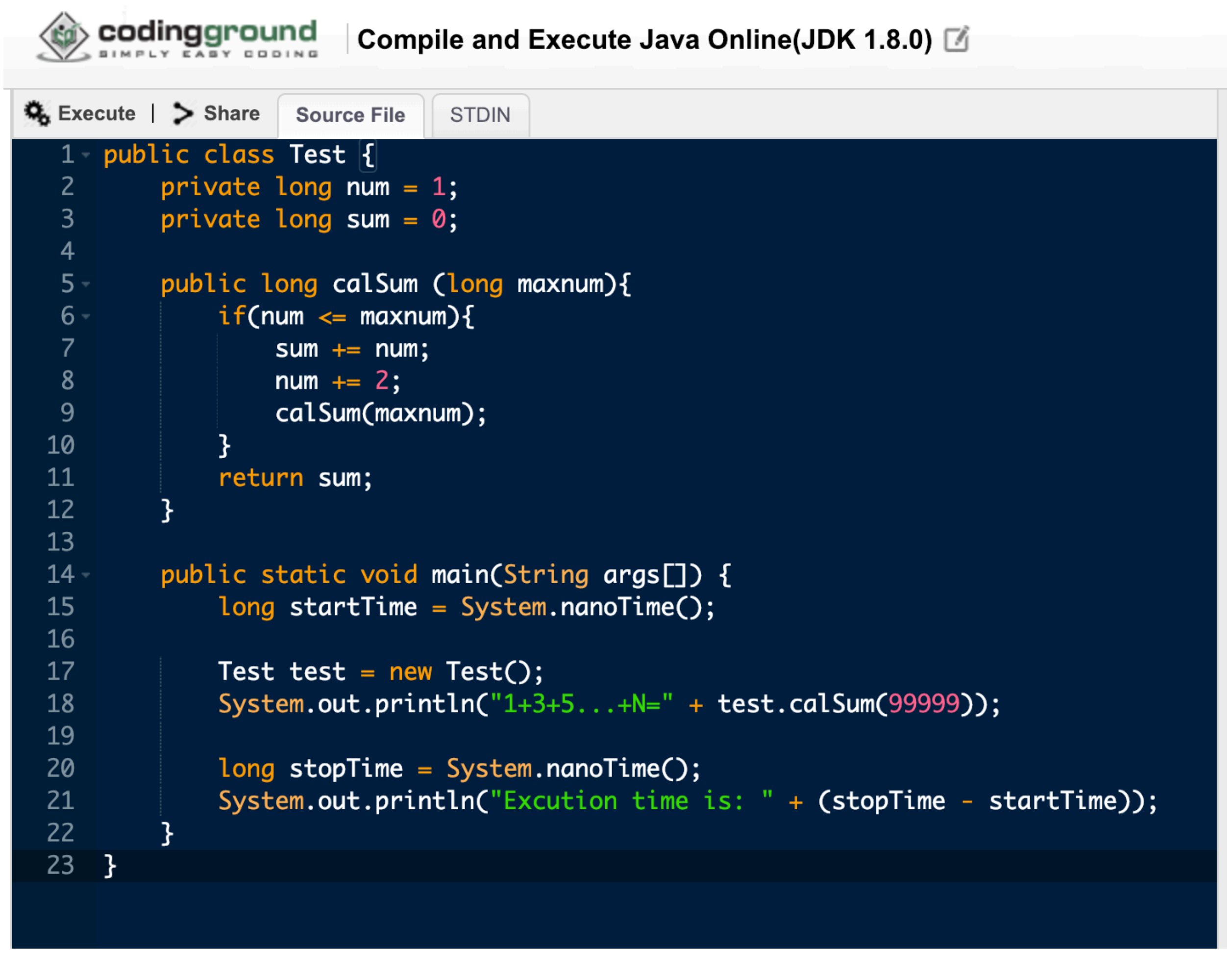
http://npu85.npu.edu/~henry/npu/classes/algorithm/growth_of_func/slide/exercise_growth_of_func.html
Q3 ==> Java program space complexity (method and static variable)
Q4 ==> C Program space complexity

Q2: Summation of N odd numbers

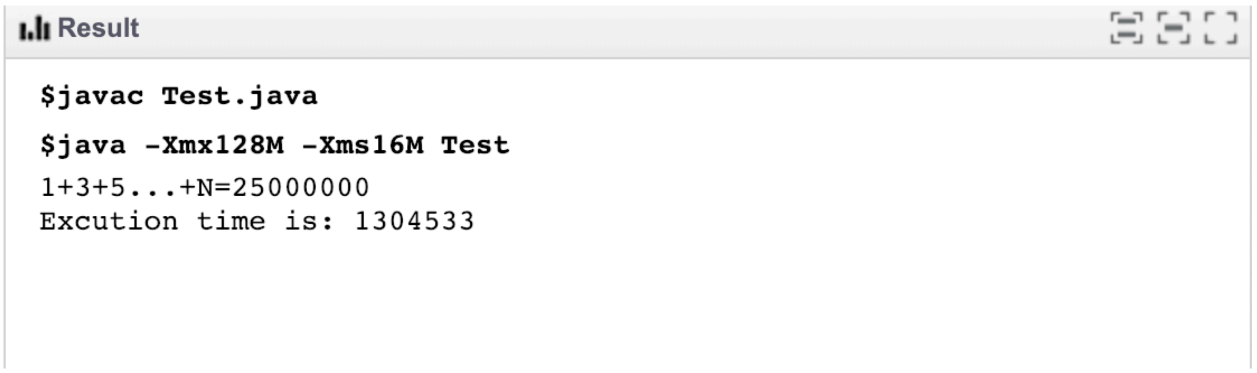
- Create two Java programs to calcultae the summation of N odd numbers. One is a recursive version, the other is an iterative version.
- Please compare the performance of these two programs by calculating

  -
    ```
    9999 + 99997 + .... + 3 + 1
    ```

- **recursive version**

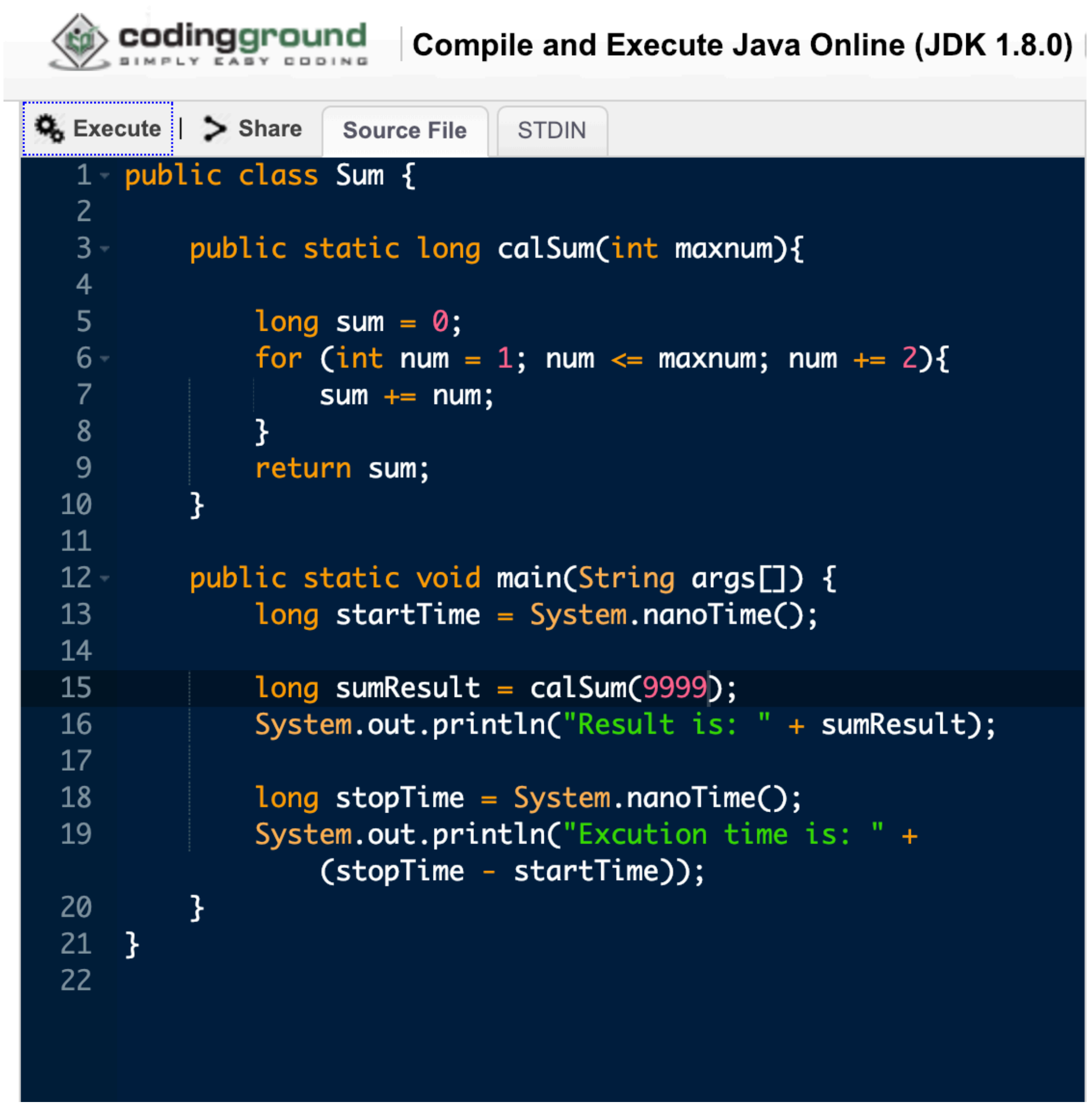


```java
public class Test {
    private long num = 1;
    private long sum = 0;

    public long calSum (long maxnum){
        if(num <= maxnum){
            sum += num;
            num += 2;
            calSum(maxnum);
        }
        return sum;
    }

    public static void main(String args[]) {
        long startTime = System.nanoTime();

        Test test = new Test();
        System.out.println("1+3+5...+N=" + test.calSum(99999));

        long stopTime = System.nanoTime();
        System.out.println("Excution time is: " + (stopTime - startTime));
    }
}
```

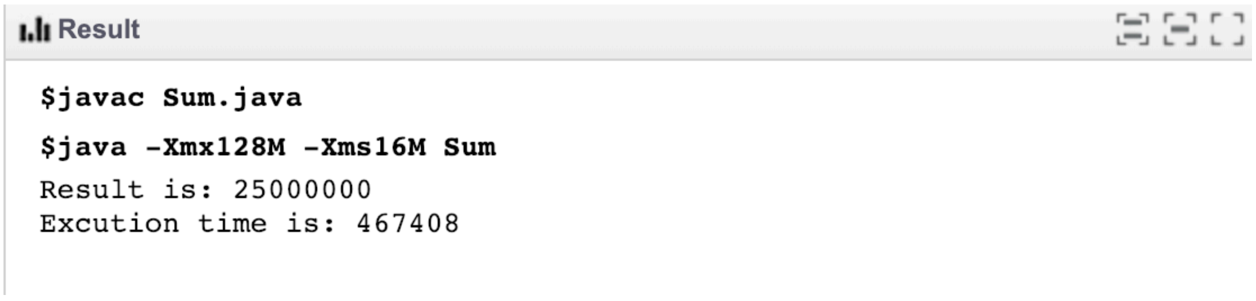


```
$javac Test.java
$java -Xmx128M -Xms16M Test
1+3+5...+N=25000000
Excution time is: 1304533
```

- **iterative version**



```java
public class Sum {

    public static long calSum(int maxnum){

        long sum = 0;
        for (int num = 1; num <= maxnum; num += 2){
            sum += num;
        }
        return sum;
    }

    public static void main(String args[]) {
        long startTime = System.nanoTime();

        long sumResult = calSum(9999);
        System.out.println("Result is: " + sumResult);

        long stopTime = System.nanoTime();
        System.out.println("Excution time is: " +
            (stopTime - startTime));
    }
}

```



```
$javac Sum.java
$java -Xmx128M -Xms16M Sum
Result is: 25000000
Excution time is: 467408
```

Q3: Java program space complexity (method and static variable) .

```
public class Test
{
    static int x = 11;                                    4
    private int y = 33;                                   4
    public void method1(int x)                                     12 + 4 =16
    {
        Test t = new Test();                                          16+ 8= 24
        this.x = 22;
        y = 44;                                                         24+4 =28

        System.out.println("Test.x: " + Test.x);
        System.out.println("t.x: " + t.x);
        System.out.println("t.y: " + t.y);
        System.out.println("y: " + y);
    }

    public static void main(String args[])
    {
        Test t = new Test();                                    8 + 4 =12
        t.method1(5);
    }
}
```

Questions:

- Please determine the bytecode size of the following Java program
- Please calculate the minimum RAM size required to run this Java program.

Answer: 28 Bytes.

Q4 :C Program space complexity

```
int i =2;                   4
void f(int j) {             5 + 4 =9
  j = i + 2;
}
void main()
{
   int k = 3;               4
   static c = '4';          1.

   {
      int m = i;
   }
   f(k);
}
```

Questions

- Please determine the executable code size
- Please calculate the minimum RAM size required to run this C program on a 32-bit machine.

Answer: 9 Bytes.