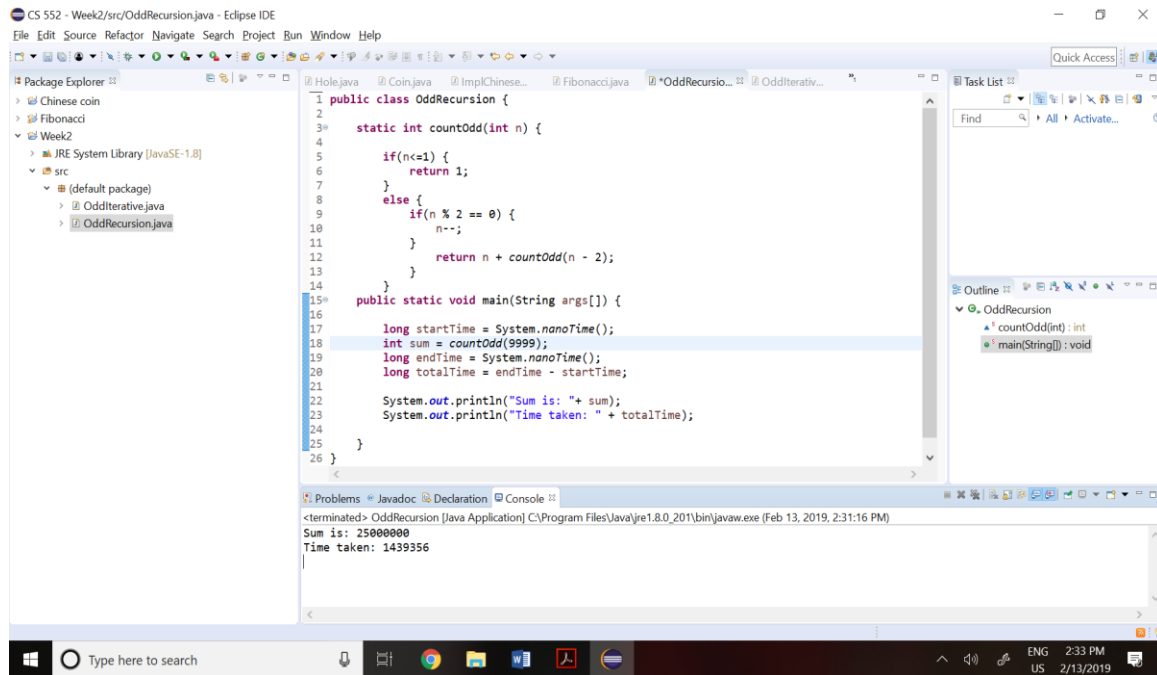


## Q.2 Summation of N odd numbers

### 1. Sum of N Odd Number (Recursion):



Code:

```
public class OddRecursion {

    static int countOdd(int n) {

        if(n<=1) {
            return 1;
        }
        else {
            if(n % 2 == 0) {
                n--;
            }
            return n + countOdd(n - 2);
        }
    }

    public static void main(String args[]) {

        long startTime = System.nanoTime();
        int sum = countOdd(9999);
        long endTime = System.nanoTime();
        long totalTime = endTime - startTime;

        System.out.println("Sum is: " + sum);
        System.out.println("Time taken: " + totalTime);
    }
}
```

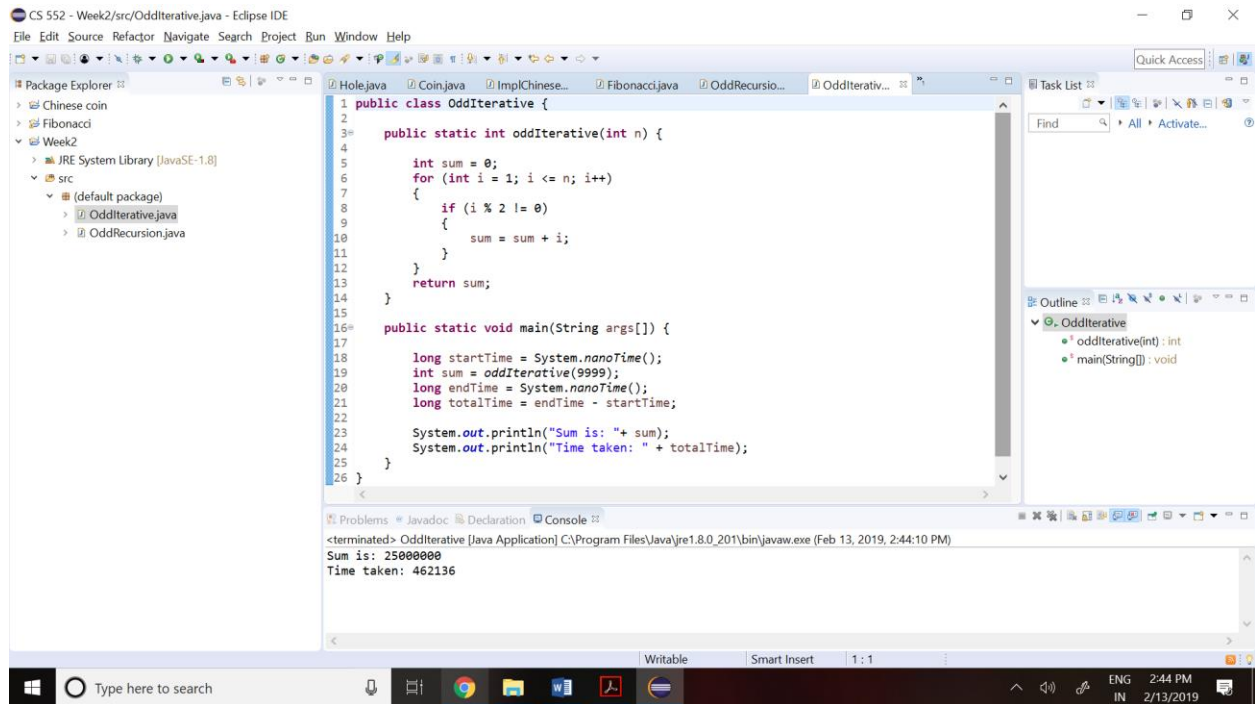
```
}  
}
```

## Result:

Sum is: 25000000

Time taken: 1439356

## 2. Sum of N Odd Number (Iterative):



## Code:

```
public class OddIterative {  
  
    public static int oddIterative(int n) {  
  
        int sum = 0;  
        for (int i = 1; i <= n; i++)  
        {  
            if (i % 2 != 0)  
            {  
                sum = sum + i;  
            }  
        }  
        return sum;  
    }  
}
```

```

    public static void main(String args[]) {

        long startTime = System.nanoTime();
        int sum = oddIterative(9999);
        long endTime = System.nanoTime();
        long totalTime = endTime - startTime;

        System.out.println("Sum is: " + sum);
        System.out.println("Time taken: " + totalTime);

    }
}

```

### Result:

Sum is: 25000000  
Time taken: 462136

### Q.3 Calculate Factorization

Java program space complexity (method and static variable) .

```

public class Test
{
    static int x = 11;
    private int y = 33;
    public void method1(int x)
    {
        Test t = new Test();
        this.x = 22;
        y = 44;

        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();
        t.method1(5);
    }
}

```

4 byte  
4 byte (Total : 8 byte)  
4 byte (Total : 12 byte)

**Maximum byte Used by the program: 12 byte**

#### Q.4 C Program space complexity

int i = 2;	4 byte
void f(int j) {	4 byte (Total : 8 byte)
j = i + 2;	
}	(Total: 4 byte ) // Scope End for J
void main()	
{	
int k = 3;	4 byte (Total : 8 byte)
static c = '4';	1 byte (Total : 9 byte)
{	
int m = i;	4 byte (Total : 13 byte)
}	
f(k);	
}	

**Maximum byte Used by the program: 13 byte**